PCM™

FEBRUARY 1984 $3
VOL.1 NO.8

THE PORTABLE COMPUTING MAGAZINE

FOR THE TRS-80 MODEL 100®
FROM THE PUBLISHERS OF THE RAINBOW

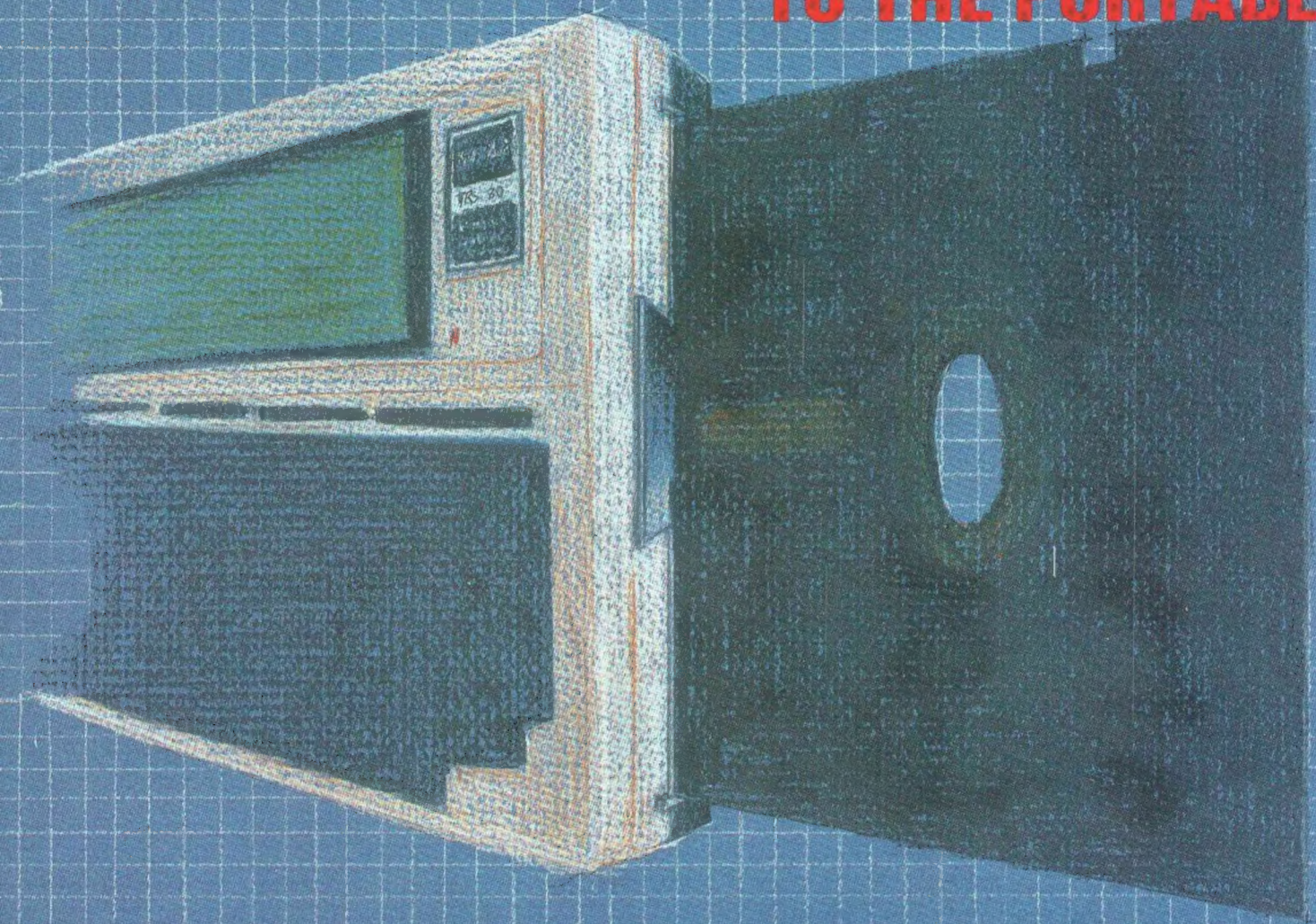
PLANNING YOUR CAR'S MAINTENANCE

MAKE YOUR RAM FILES INVISIBLE

DATA HANDLING TECHNIQUES

SIX FINANCIAL PROGRAMS IN ONE

MASS STORAGE COMES TO THE PORTABLE



02
0 44254 00002

From the people who brought you... BUSINESSPAK+

Five NEW programs designed for the businessperson.

Each on its own cassette with interesting manual.

First with software for the Model 100

You have made BUSINESSPAK+ an overwhelming success! Now, we are introducing **five new programs** for the Model 100.

PCSG continues to be first in innovative programming for the businessperson using the TRS-80 Model 100 Computer.

Look at these five new programs.

DATA+ $59.95

With DATA+ the Model 100 becomes virtually a true data base. You can input data rapidly from a 16 field screen that you make yourself like our PUT+. Here is the difference:

REVIEW / Using function key 1 you can recall any record, by searching any field.

EDIT / Change or update any record you select, instantly on the screen.

LIST / Using function key 5 you can print address labels, list records or selected fields of records in columns or other configurations. You can even pause, and reset left margins.

MERGE / Function key 6 lets you merge. You can automatically print any fields of any records into forms or letters, wherever you designate. With all four of these functions you have full search and selection capability.

With **LIST** and **MERGE** DATA+ remembers your favorite formats, quickly defaulting to them by simply pressing the ENTER key. The added feature **BUILD** lets you build and print a file of unrelated records that could not be selected either alphabetically or numerically.

On cassette with excellent, easy to understand manual.

TYPE+ $59.95

The Model 100 now becomes, with the addition of any printer, the finest, most feature rich, electronic typewriter available today. You can type directly to the paper. (Some printers have a one line buffer.)

SCREEN BUFFER / To allow you to edit before printing, you can control the screen buffer from 1 character to the last line.

CENTER / Center on / center off controlled with function keys.

MARGINS / Set margins with function keys. Audible end of carriage, automatic carriage return and function key Tab Set.

DIRECT CONTROL / If your printer responds to backspace commands, you can backspace and overstrike. Paper advances with carriage return.

With TYPE+, everything you print is stored simultaneously in a RAM file, formatted as you composed it. You can reprint instantly, or edit it in the file. On cassette with excellent, easy to understand manual.

TUTOR+ $29.95

Learn keyboard skills by playing a delightfully exciting game. While experiencing the thrills of a space invaders type game you become proficient at manipulating the keyboard. A game that forces you to learn touch typing, utilizing all the keys. On cassette with excellent, easy to understand manual.

PCSG provides hotline software support for the Model 100. Give us a call at 1-214-351-0564.

SORT2+ $29.95

You can sort a file in place with SORT2+. It consumes only 1k free memory, while sorting file in RAM you entered with PUT+ or DATA+. Our original SORT+ allows you to sort from cassette, but requires more memory while sorting. SORT2+ is for those times when memory or cassette sorting is a problem. SORT2+ also has upper case fold, and true numeric field sort. On cassette with excellent, easy to understand manual.

TENKY+ $59.95

With TENKY+ the numeric keypad on your Model 100 emulates both a ten-key desk calculator, and a financial calculator. The right side of the screen represents the tape, which can also be directed to a file and/or a printer. Ten working registers are visible on the left of the screen, each one functioning independently as a calculator with functions including exponentiation and logarithms. Registers are accessible from each other, including the tape which stores up to 120 values, each associated with a note of up to 26 characters.

Financial functions available at the touch of a key include IRR, MIRR, and NPV on up to 120 cash flows, plus any of its six annuity variables (n, PV, FV, PMT, i). Function key F2 gives a printed amortization schedule, and businessmen will love the breakeven analysis they can receive simply by pressing function key F1.

Portable Computer Support Group is pleased to offer these program additions. We endeavor to continue as *The Leaders in Software for the Model 100.*

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd. No. 207, Dallas, TX 75229

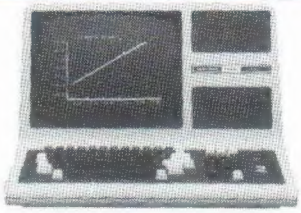
MASTERCARD / VISA / COD

From Computer Plus to YOU . . .

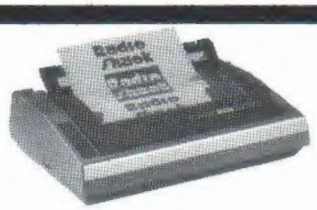
PLUS *after* PLUS *after* PLUS

Model 100 8K $679
Model 100 24K $835

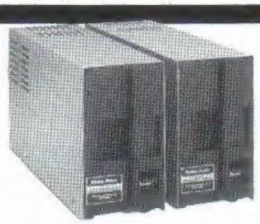
Color Computer II 16K $135
w/16K Ext. Basic $165
w/64K Ext. Basic $210



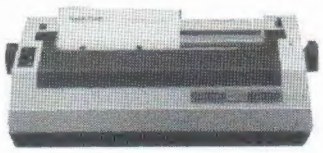
Model 4 16K $849
Model 4 64K
2 Disk & RS232 $1699



DMP120 $395
DMP200 $520



Color Computer Disk Drive
Drive 0 $329 Drive 1 $235



DWP210 $629

BIG SAVINGS ON A FULL COMPLEMENT OF RADIO SHACK COMPUTER PRODUCTS

COMPUTERS	
Model 4 Portable 64K 2 Drives	1525
Pocket Computer 2	165
Pocket Computer 3	89
Pocket Computer 4	59
Model 2000 2 Drive	2299
Model 2000 W/Hard Disk	3599
Model 12 64K 1 Drive	2699
Model 12 64K 2 Drive	3375
Model 16B 1Dr 256K	4249
Model 16B 2DR 256K	4915
MODEMS	
Hayes Smartmodem II	225
Hayes Smartmodem 1200	539
Novation Smartcat 1200	410
Novation J-Cat	115
AC-3	129
DC Modem I	89
DC Modem II	160

PRINTERS	
Silver Reed EXP500 D.W.	425
Silver Reed EXP550 D.W.	575
Daisy Wheel II	1745
DWP410	1159
DWP210	629
CGP115	159
CGP220 Ink Jet	545
DMP110	305
DMP420	735
DMP500	1219
DMP2100	1689
Gemini 10X	300
Delta-10	470
Gemini 15X	405
Prowriter	379
Prowriter II	649
Okidata	CALL
Epson	CALL
DISK DRIVES	
Model III/4 Drive 0	515
Model III/4 Drive 1	195
Model III/4 Drive 2	289
Model III/4 Drive 3	259
Primary Hard Disk M12	2689
Primary Hard Disk MIII	1799
ETC.	
CCR-81 Recorder	52
Cassette Tapes (10 Pk)	9.95
16K Ram Chips	25
64K Ram Chips	75
Model 4 64K Upgrade Kit	140
M100 Bar Code Reader	89
64K Printer Buffer	199
Printer Stand	25
Printer Cables	
Printer Ribbons	
Printer Paper	
Dust Covers	
Computer Books	
Radio Shack software 10% off.	
Send for complete listing of brand name software and hardware.	

CALL TOLL FREE
1-800-343-8124

- *LOWEST POSSIBLE PRICES*
- *BEST POSSIBLE WARRANTY*
- *KNOWLEDGEABLE SALES STAFF*
- *TIMELY DELIVERY*
- *SHOPPING CONVENIENCE*

VISA AMERICAN EXPRESS MasterCard

computer plus

P.O. Box 1094
480 King Street
Littleton, MA 01460

SINCE 1973

IN MASSACHUSETTS CALL (617) 486-3193

TRS-80 is a registered trademark of Tandy Corp.

Contents

PCM

The Portable Computing Magazine

Vol. 1, Issue 8 February 1984

Editor and Publisher
Lawrence C. Falk

Managing Editor Courtney Noe
Senior Editor Jim Reed
Assistant Editor Susan A. Remini
Submissions Editor Jutta Kapfhammer
Editorial Assistants Valarie Edwards, Wendy Falk, Lynn Miller, Shirley Morgan, Noreen Morrison, Kevin Nickols
Contributing Editors Dan Downard, Robert Frowenfeld, Richard White
Art Director Jerry McKiernan
Assistant Art Director Sally Nichols
Designers Peggy Henry, Neal C. Lauron
Advertising Manager Charlotte Ford
Advertising Assistant Lynda Wilson (502)228-4492
General Manager Patricia H. Hirsch
Assistant General Manager for Finance Donna Shuck
Bookkeeper Diane Moore
Administrative Assistant To The Publisher Marianne Booth
Customer Service Manager Suzanne Kurowsky
Assistant Customer Service Manager Deidra Henry
Research Assistants Monica Wheat, Wanda Perry
Dispatch Mark Herndon

PCM is represented in the Eastern United States by Garland Associates, Inc., P.O. Box 314, S.H.S., Duxbury, MA 02331, (617) 934-6464 or 934-6546. Advertisers east of the Mississippi may contact them for further information. West coast advertising representative — Cindy Shackleford.

PCM—The Portable Computing Magazine is published every month of the year by FALSOFT, INC., 9529 U.S. Highway 42, P.O. Box 209, Prospect, KY, 40059. Phone (502) 228-4492. **PCM — The Portable Computing Magazine** and the PCM logotypes are ® Trademarks of FALSOFT, Inc.

Second class postage pending at Prospect, KY and additional offices. POSTMASTER: Send address changes to PCM, P.O. Box 209, Prospect, KY 40059. Forwarding Postage Guaranteed.

Subscriptions to **PCM — The Portable Computing Magazine** are $28 per year in the United States. Canadian and Mexican rates are U.S. $35. Surface mail to other countries is U.S. $64, air mail U.S. $85. All subscriptions begin with the next available issue.

Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in United States currency only.

Limited back issues are available. Please see notice for issues which are in print and costs. Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in U.S. currency only.

Features

Spelling Practice/ *Mel Perkins* 11
Incorporate voice into this educational tool

More Storage For PoCo/ *Jim Hawk* 16
Hungry for Storage? Grab a megabyte to go!

Game For One/ *Bob Delbourgo* 22
Try this version of solitaire

The Loan Arranger And PoCo/ *William T. Stauffer* ... 24
Hire your silver with these helpful routines

Invisible Files/ *Ronald Paludan* 29
Keeping secrets in RAM

Variable Lister/ *Larry Randall* 30
Help your computer become a programming partner

Two Helpful Hints/ *Willis Rollins* 32
Handy information you may have overlooked

Text Mate/ *Tino Delbourgo* 34
This text utility performs three useful functions

Departments

BASIC Bytes/ *Richard White* 8
Data handling techniques

Letters To *PCM* 6

LPRINT/ *Lawrence C. Falk* 5
Editor's Notes

New Products 37

On The Road/ *Robert Frowenfeld* 12
Maintenance schedule for your car

Product Reviews 36

The Rackseller 38

Advertisers Index

BT Enterprises **18**
Chattanooga Choo Choo ... **6**
Computer Plus **3**
Computer Solutions **36**
Diskus Products **37**
Federal Hill Software **35**
Leading Edge ..... **BC**
PCM **19**
Portable Computer Support Group ..... **IFC**
Prickly-Pear Software **7**
Purple Computing **28**
Radio Shack **20-21**
Rainbowfest **25**
Silverware **17**
Skyline **33**
Spectrum Projects ..... **IBC**
York 10 ..... **31**

Cover by Jerry McKiernan © 1984 by Falsoft, Inc.

'Peripheral Vision' — A Look At Third-Party Support

Many of you have, at this point in the game, had your Model 100s for close to a year now. It seems to me that you've probably developed some pretty good idea of the things you like and dislike about it and likely have developed some specific thoughts of your own on things you would like to see incorporated into this powerful, lightweight machine to make it even more personally useful than it already is.

Since we have to allow for such a wide range of individual preferences and, at the same time, maintain portability, it seems most probable that any additional features might best be in peripheral form — such as, say, an external color monitor interface. And that's already available in the Mikrokolor Color Graphics Interface from Andreasen's Electronics as listed in our New Products section last month.

How about a rechargeable battery system? After all, those "AA" alkaline cells aren't free and our friendly PoCo seems to have quite an appetite for them. Or, maybe instead of balancing our computers on our laps while winging our way to the coast we could have, say, a carrying case that would open up and serve as a sort of portable desk?

Well, so far, it seems that every time we come up with a need someone not only has anticipated that need but filled it, because you'll find in this issue's New Products section both a rechargeable battery system and a combination carrying case and lap-desk already for sale — just for your Model 100.

What's that? You've caught me backing into this discussion of peripherals and needs by peeking into the *PCM* files for examples? Well, that's true, but the idea I want to convey here is that while we've been using our portables and discovering areas of need, others have discovered those same areas, not as a lacking in the Model 100, but as golden opportunities to strengthen this computer and to make it even more powerful than before. As we have found in publishing *the Rainbow, PCM's* sister publication for the TRS-80 Color Computer, third-party support for a computer can bring power and sophistication that was undreamed of when the machine first came on the market. And, by strengthening the computer, these companies are directly responsible for increased popularity and sales of that computer and, thereby, increase their own market base and make entry into that market by other potential support groups more likely. The word "symbiosis" keeps coming to mind.

Other improvements? How about a device which will enhance a peripheral you already own, such as a printer? I'm thinking of a means to allow your tractor-feed printer to use ordinary stationery. That's reviewed in this issue, too.

Well, then, how about mass storage? It's our featured article — we've got you covered!

— Lonnie Falk

Letters

See PCM 4/84 p. 6

Editor:

There are some goodies stored in PoCo's ROM which seem to be known by professional programmers, but which are not mentioned in the manual. Those I've found so far include @, ABS(X), ASC(X), CARAT (shifted 6), and MOD.

@, followed by a number greater than 40 and a comma, permits printing at an indented position on the line.

ABS(X) returns the absolute value of X, removing the negative sign, if any.

ASC(X) returns the decimal equivalent of any ASCII character that can be produced from the keyboard, including those requiring GRPH and CODE; but it doesn't work with comma, quotation mark, space bar, or enter, possibly others.

The carat introduces an exponent, permitting the user to raise a number to any power. For a power of 2, however, it is faster to multiply the number by itself, i.e., Y=X*X rather than Y=X^2.

MOD is modulo. It returns the remainder after one number is divided by another. Thus 13MOD5 returns a 3. (MOD will ignore any digits after the decimal point in either of the numbers).

I've seen the backslash (\), produced by holding down GRPH and keying the minus sign, used in a commercial program; but I haven't figured out what it does.

Compared with most other printed documentation for computers and software, the manual is not bad; but it is not nearly as good as the machine. David A. Lien's book "The TRS-80 Model 100 Portable Computer" is an excellent tutorial for someone at my level of ignorance, but it also leaves some gaps.

A. W. Goldman
Newton, MA

Editor:

My wife and I travel by motor home in our job and have found the TRS-80 Model 100 to be a very effective tool for handling correspondence, reports and schedules. In a motor home, storage space is limited, and while the Model 100 posed no problem, we also had to carry a printer and portable typewriter. We have now eliminated part of the problem by purchasing a Brother portable typewriter which can be interfaced to the Portable 100. It works fine for our purposes and we can leave the printer at home when we travel.

William S. Auman
Bella Vista, CA

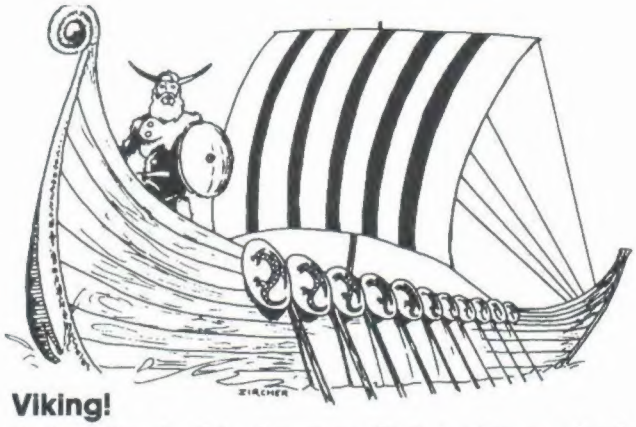
PRICKLY-PEAR SOFTWARE
QUALITY PROGRAMS FOR YOUR MODEL 100

Monsters & Magic

The most realistic fantasy role-playing game yet for the Color Computer. You start out by rolling up your character's basic ability scores and buying equipment. When you are ready it's into the dungeon. Your character starts at first level, but can rise in levels by garnering experience in the fray. If you have role-played fantasy games you will be amazed at the realism of the combat system. Armor class, initiative, and damage by weapon type are all included, with over 50 different monsters to fight - each with it's own abilities. As you rise in level you can win treasure and find magic weapons and spells - if you live!! You set the game length by telling how many monsters you want to fight before you reach your final battle to the death against the powerful Dungeon Lord. There are 1000+ place description combinations in this text based game, and real excitement in every one! This is a fantasy simulation, and is truly not like any adventure game you have ever seen. For 1 player. **$19.95 tape—24K**

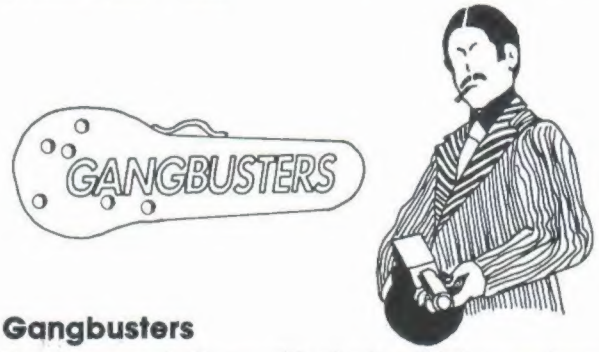
Astrology

Truly a classic, this program will accurately cast your complete horoscope. You just enter the date, time, and place of birth. The sun sign, rising sign, mid heaven (MC), lunar nodes, and planetary influences including houses and aspects between the planets will all be calculated, and a full chart drawn. You can also do progressed charts and transits. It will even tell you the day of the week you were born. The accompanying book will help you interpret this chart of your horoscope. The extent of the documentation is tremendous, even by our exceptionally high standards, and no previous knowledge of the subject is required. You can share in this wisdom which has been used for thousands of years in many cultures. This program was written by a professional Astrologer. **$34.95 tape — 16K**



Viking!

A simulation for 1 to 4 persons. Each begins as a landowner, and by farming their land, buying and selling land, expanding their fishing fleet, building on to their manufactory, increasing their population, equiping and training more soldiers, and regulating their taxes, each player tries to increase their economic power and rank until one becomes ruler over all. But beware plagues, rats, raiders, revolts, bad weather, and other misfortunes which may lie along the road to success. As you progress, see the map of your holdings increase. Playable in 1 to 2 hours, and different every time, you may have an addiction problem. **$19.95 tape — 24K**



Gangbusters

If you ever wanted to try a life of crime, this is your chance. You will start out as a Punk, but by using brains, and a little muscle, you can rise to become a Hood, Runner, Bookie, Torpedo, Fence, Kingpin, or win by becoming Syndicate Boss. Indulge yourself. Bribe a judge, or the District Attorney. Pay off the Cops. Take out a contract on another player, but watch out, they may be after you. Buy trucking companies, bootleg operations, houses of ill fame, but remember, if you get caught, you may do some hard time. Do you have what it takes to take over? This game will keep you close to your rod, get you thinking about bulletproof glass in your car, and definitely bring out the worst in you, but you'll love every minute of it. For 2 to 6 players, takes about 2 hours to play. Every game is excitingly different. **$19.95 tape — 24K**

Your personal check is welcome - no delay. Include $1.50 shipping for each program ordered. (Shipping free on $50.00 or larger orders). AZ residents add 7% sales tax. Orders shipped within two days.

Stocked by Quality Dealers, or

Send Order To: **PRICKLY-PEAR SOFTWARE**
9234 E. 30th Street
Tucson, Arizona 85710
(602) 886-1505

MasterCard VISA

Techniques For Handling Data

By Richard A. White
PCM Contributing Editor

In *PCM's* sister publication, *the Rainbow,* there is soon to be published an article by Professor Sam Sherrill on using CoCo and the M100 to conduct surveys and to analyze the results. Because fairly large quantities of data are involved, both speed and memory conservation are important when writing programs for use in survey research. As Professor Sherrill points out, surveys are conducted by asking a sample of people a standardized set of questions and recording the results, usually on paper questionnaires. The largest survey organizations use some form of CATI, computer-assisted telephone interviewing. CATI systems allow interviewers to directly enter answer codes into a minicomputer or mainframe terminal. However, such equipment is expensive and clearly not feasible for small organizations that only occasionally conduct surveys among clients or customers.

Professor Sherrill wondered if he could use his CoCo to help with a 50-question, 100-person survey. He wrote a BASIC questionnaire program and then trained interviewers to conduct interviews over the telephone using CoCos he supplied. It worked! Each question and its accompanying responses appears separately on the screen for the interviewer to read to the respondent. The answers are assigned codes between 1 and 10 (not all numbers need be used for a question). After each response was obtained, the appropriate numerical code was keyed in. The next question then would be written on the screen. At the end of the interview, the interview number and the response numbers were saved to a separate disk file. This worked quite well in getting the data.

The portability of the M100 adds a new dimension to microcomputer surveying in that the machine can be taken with an interviewer doing face to face personal interviews. Data files can be easily backed up to tape during the day. At the end of the day, data could be dumped over the phone to a base computer, which could be anything including another M100. Data analysis programs that run on the CoCo can run on the M100 as well. The only real lack is that there is no disk drive, which slows the operation somewhat. Memory constraints are similar. A small business could run a small survey of its customers and do all the analysis on one M100.

Before I get anyone's hopes too high, I have to say that there would need be some programming tailored to the particular survey. Further, survey design is not easy and the services of a consultant specializing in the field would be very useful in developing a questionnaire that produces useful results. So, the rest of this column will deal with some programming techniques that we used to handle the survey data and which can be profitably used for other data handling projects. The survey example provides a real life application.

Simple saving and loading data from tape is not particularly difficult. The descriptions in the M100 manual give the user a good start. However, when there are other considerations, such as the most efficient storage of data in the computer, things get somewhat more difficult. The data from one interview in Professor Sherrill's survey consisted of a file of 45 single-digit numbers. (These files were on disk, but could have been on tape as easily.) Since there would be over 100 interviews, a program to calculate and display the results of the survey would have to process more than 4500 numbers. If two bytes are needed to store each number, as is the case with an integer, over 9000 bytes would be needed for data alone. This is tight in a 24K machine which may need to store other files or deal with larger databases. Fortunately, there is a way to handle this much data (and more).

Numbers can be stored very efficiently in strings. This is done with the MID$ function to store a character or string in another string. This is one of those BASIC commands that pays for itself many times over. In the example above, storage requirements are cut in half. Impressive! And not particularly difficult to do. This means that a 32K M100 can handle perhaps 300 interviews of 50 questions each in memory with the analysis program.

Suppose we have fifty numbers per interview or record and that each is a single digit number. How are the numbers stored now? Well, they could be stored in an array like V(1), V(2) . . . V(50). You would have dimensioned the integer array with a statement like DIM

Dick White has been programming in BASIC *for over three years, and has a number of programs on the market for the Color Computer. He is also a columnist for* ***the Rainbow, PCM's*** *sister publication, and is a member of CINTUG, the Cincinnati TRS-80 Users' Group.*

V%(50) early in the program before any numbers were entered. What will the final string have to look like? We need to know this before we decide to put the string together because there are a number of options. First, will there be string identification in the string itself? Good idea to have it. Next, what kind of identification will this be? It could be simply a number or something more complex like a name or other sequence of characters that could vary in length from string to string. Keeping things simple for now, we will use a three digit number to identify the string, and will put it in the first three positions in the string.

Where does this string identifying number come from? It could be entered from the keyboard or it might be assigned by the program itself. In either case, BASIC code will have been written to get the identifier into a variable that is available when the data string is to be assembled. In our survey example, the interviewer entered the interview number in response to the first question in the interview: this number is the first number in the array in V(1).

```
30 A$=STR$(V(1)):IF  LEN(A$)>3
   THEN A$=RIGHT$(A$,3)
35 IF LEN(A$)<3 THEN A$=" "+A$
```

To put the number into a string we must convert it to a character from its numeric form. A$=STR$(V(1)) does this. A$ could be four characters long since STR$ always adds either a leading space or a minus sign when making the conversion. We can test this: if A$ is four characters long, get the right three with A$=RIGHT$(A$,3). On the other hand, our number could be a single digit and A$ would be only two characters long. We can fix this with A$=" "+A$, if necessary. We now have A$ string that is exactly three characters long and which may contain any number from 0 to 999.

Next we need to put A$ into the data string. Easily done. If we are calling our data string ST$(X), ST$(X)=A$ does the job, assuming that ST$(X)=" " before the assignment. If ST$="SOME SORT OF GARBAGE" there is trouble in River City, since our method of decoding the string will depend on a particular piece of data being at a particular location in the string.

```
50 FORY=2TO45:  A$=SRT$(A$):
   A$=RIGHT$(A$,1): ST$(X)=ST$
   (X)+A$: NEXT
```

In the above line, a simple FOR . . . TO . . . NEXT loop is run to get all the rest of the values, convert them to single character strings, and add each in turn to the the data string ST$(X). For those with a fancy for large words, the string addition process is called concatenation. Note that we had to get the right character in A$ after the STR$(V(Y)) operation to make A$ one digit long.

There is a better way to put data into strings. Any time a string is redefined, the new string is written to a new location in string memory and the location of its old contents is forgotten. The old string is still there wasting memory. Eventually, string space fills up with garbage strings and M100 must stop to toss the trash. Fittingly, the process is called garbage collection. In most programs, this hesitation is no problem. However, when string space is tight, or there is a whole bunch of string processing going on, garbage collection can seriously delay a program. Here is a rewritten code that chucks concatenation and uses MID$ "on the left."

> "The portability of the M100 adds a new dimension to microcomputer surveying in that the machine can be taken with an interviewer doing face to face personal interviews."

```
25 . . . :ST$(X)=STRING$(50," ")
40 MID$(ST$(X),1,3)=A$
50 FORY=2TO45:  A$=RIGHT$
   (STR$(V(Y)),1): MID$(ST$(X),
   Y+2,1)=A$: NEXT
```

The last statement in line 25 defines ST$(X) to be a string containing 50 spaces. We cannot now add our string identifying number to a null ST$, but in line 40 we can insert it into the first three positions in ST$(X) in place of the blanks. Actually, we could have filled the string initially with any character. If we had used a period to fill ST$(X) the string would now look like this.

```
145 . . . . . .
```

The 145 is a string identifying number or record number inserted by the code in line 40. In line 50, the process of defining and processing A$ is unchanged. What is different is that MID$(ST$(X), Y+2,1)=A$ inserts the character into a calculated location in the string. The string is never rewritten in string space and little garbage is collected. The same process can be used in an editing program to change data in the string.

MID$ statements are a bit tricky to deal with since you must calculate where in the string data goes and write BASIC statements to implement these calculations. On the other hand, the technique can give you more control over your data and provide memory economy. More important, this idea of inserting data in a certain point in a record or in memory is central to data handling in assembly language, C, PL/1, PASCAL and BASIC09 and other applications where fixed memory space can or must be allocated. The more we ask a language to manage our data, the higher the price in memory usage and speed. Where neither matters, the strengths of BASIC should to be used to minimize programming time.

For Professor Sherrill's survey, a short program was written to take numbers from disks and convert them to the string format. These files of strings were then run into another short program that combined them into a single file. This final file is the survey database. In the M100 environment and in future CoCo surveys, I would expect to do the string filling during the interviewing. To create the database, a file of numbered empty strings was generated. This file held more strings than we expected to need. The combining program loaded the full string file and then entered each small file and overwrote the empty strings. This allowed us to write the editing and tabulation programs and use them on a partial database before the survey was completed.

Two types of basic data tabulations are made in survey analysis: frequency counts (often called straight runs and marginals) and cross-tabulations. Straight runs are absolute frequency distributions while marginals are frequency distributions expressed in percentages. Cross-tabulations (one-way) are the answers to one question partitioned by the answers to a second question. In a sample survey of M100 owners, one question we would certainly ask is

whether the respondents read *PCM*. A straight run on this question would yield the number who answered "yes" and the number who answered "no": the marginal would yield the percentages answering each way. We might also ask the respondents to our survey where they buy software and accessories. We wonder whether respondents who read *PCM* have different buying patterns from those who don't. For example, do they buy more by mail order and deal less with local sources, including Radio Shack? We can find out by cross-tabulating the answers to the buying question by the answers to the readership question, as follows:

	Read PCM	
Buy from	Yes	No
Radio Shack only	How many	How many
Local non-R.S. store	How many	How many
Mail from multi-line vendor	How many	How many
Mail from publisher or manufacturer	How many	How many

The data in the table above could be kept in a two-dimensional array in the M100. The minimum array would be DIM NM%(2,4). I visualize that as two columns of four rows each. In the cross-tabulation program, I defined an array as DIM DA%(11,11). This is actually a 12 by 12 array since there is a zero row and a zero column. A bit over 290 bytes of storage are needed with integers.

Actually, this array was used in the second try at the program and there is a lesson in that story. Professor Sherrill wrote the first cross-tabulation program, and we only added the code necessary to enter the file of strings from disk and decode them. It was at that point that large, random-number data blocks were processed for the first time and the program was found to be slow. Professor Sherrill had worked out a way to calculate just where in a one-dimension array a particular number should be stored. The calculation worked, but took time. I was asked to investigate a machine language approach. In thinking about the task, I visualized a data table that felt like a BASIC two-dimensioned array. I rewrote the program first in BASIC and we found it to be over 10 times faster. The machine language program was never written.

I certainly had not expected to make this type of impact. The lessons from this effort are simple and important: calculations take time and where they can be reduced or simplified, program running time is reduced. Division and raising numbers to powers are the slowest calculations. Addition and subtraction are fastest. Multiplication is in the middle. When you have the choice of making a calculation before entering a loop or inside the loop, always do it before. The heart of the cross-tabulation code is below. Note that only tests (IF . . . THEN) and addition are used.

```
470 CLS:PRINT:PRINT"INITIALIZING ARRAY":FORX=0TO
    11:FORY=0TO11:DA(X,Y)=0
    :NEXT:NEXT
480 CLS:PRINT"ENTER THE VARIABLE NUMBER OF THE IN-
    DEPENDENT(COLUMN)VARIABLE",:INPUTI:CLS
490 PRINT"INDEPENDENT VARIABLE IS"I".","ENTER THE
    CODES YOU WISH TO USE FOR THE INDEPENDENT
    VARIABLE. AFTER ALL THE CODES HAVE BEEN ENTER-
    ED, ENTER 0 TO CONTINUE."
500 INPUTX:DA(X,0)=X:IFX=0
    THEN 510 ELSE 500
510 CLS:PRINT"ENTER THE VARIABLE NUMBER OF THE DE-
    PENDENT (ROW) VARIABLE",:INPUTJ
530 CLS :PRINT"DEPENDENT VARIABLE IS"J".","ENTER
    THE CODES YOU WISH TO USE FOR THE INDEPEN-
    DENT VARIABLE. AFTER ALL OF THE CODES HAVE
    BEEN ENTERED, ENTER 0 TO CONTINUE."
540 INPUTY:DA(0,Y)=Y:IFY>0
    THEN 540
550 CLS :PRINT@256,"THE RESULTS WILL APPEAR","MO-
    MENTARILY."
```

In the above block, the user is asked to enter the question numbers ("variable numbers" to the pros) I and J to be cross-tabulated, and which answers (represented y code numbers) are to be considered. In lines 500 and 540, the codes are obtained and simply put into their numerical positions in the array i.e. DA(X,0) or DA(0,Y). All other array, members are still zero.

```
570 CT=0 :FORA=NM TO NM+
    NR-1 :GOSUB110 :IF V=DA
    (V,0)ANDV>0 GOSUB120 :IF
    V1=DA(0,V1)ANDV>0 THEN
    DA(V,V1)=DA(V,V1)+1 :DA
    (V,11)=DA(V,11)+1:CT=CT+1
580 NEXT:CLS
```

Line 570 does the real work in a elegantly simple way. The FOR . . . TO . . . NEXT loops through the entire string array. Subroutine 110 gets the answer (code) for the first question in V. If V=0 there is no data in that string and no point in even checking the second question. The same is true if the code returned was not chosen for tabulation and DA(V,0)=0. Then the test V=DA(V,0) fails. In both cases, control falls to 580 with its NEXT. If V=DA(V,0) is true, the code for the second question is obtained and a second set of tests are made. If V1=DA(0,V1), we have a match. Array members DA(V,V1) and DA(V,11) are incremented by one as is the count variable CT. DA(V,11) is a column total while CT is the number of interviews tabulated. Three simple additions and the program goes to the next pair of strings. A sample array might look like this.

```
 0  1  2  3  0  0  0  7  8  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0
 2 13  7 15  0  0  0  3  1  0  0  0
 3  8  2  2  0  0  0  4  0  0  0  0
 4  4  7 18  0  0  0  2  2  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0
 7  1  2  4  0  0  0  1  1  0  0  0
 8  2  4  0  0  0  0  0  1  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0
 0 27 22 39  0  0  0 10  5  0  0  0
```

Once the data are in the array, it can be easily printed to screen and printer. The program can check DA(V,0) and DA(0,V1) and print only rows and columns that carry values other than zero. Percentages are calculated at the time of printing, pulling data from the array, but not changing it.

The set of survey programs discussed here are limited in use to one particular survey and will need modifying before they can be broadly applied; hence, they are not published in full. In presenting code portions, I have tried to illustrate some different ways to use BASIC to meet real computing needs. Putting numerical data into strings can solve memory limitation problems, and reduce tape save and load time. Data manipulation in arrays is very powerful and fast. In combination, we have found another practical and important application of the CoCo and probably for the M100 as well.

PCM

If your spelling doesn't compute, what you need is . . .

SPELLING PRACTICE

By Mel Perkins

As I began to write programs on my Model 100 for use in my business, I also began to see its many uses as an educational tool for my son. As with many kids, he was very eager to spend time on the computer.

To help maintain his interest in using the computer, and provide him with a useful learning experience, I wrote the following program. It is a spelling practice program, and I hope it will be useful to your child (or you) as well.

Setting Up

First, edit line 10 to contain the words to be spelled, placing them within the quotes. The last entry must be "stop" to tell the program when the last word has been given. The user will not be asked to spell "stop"; however, if it is omitted, an out of data error will result. A handy way of placing the words in the data statement is to place (",") in the paste buffer, inserting it between each word.

Next, the words are to be placed on a cassette tape. It is preferable to use a leaderless tape; however, a standard leader tape may be used by inserting this line:

```
25 PRINT@176,"WAIT":MOTOR-
   ON:FORX=1TO3200: NEXTX:
   MOTOROFF:CLS
```

Run the program and type "LOAD" on the instructions display. This will take the program to line 500 at which point you must set the cassette player to RECORD and prepare to speak the first word into the microphone. Don't forget the only plug from the cassette interface cable to be plugged into the recorder is the remote. Ready? Press ENTER, speak into the microphone the first spelling word. (If your recorder doesn't have a built-in microphone, you will need to use an external microphone like Radio Shack's 33-1054 that has either separate "mic" and remote plugs or a mic plug only. The microphones that come with some recorders won't work in this application because their dual-plug connectors prevent you from plugging in the Model 100's remote plug. Disconnect the Model 100 from the AUX and EAR jacks, and plug the microphone into the MIC jack.) The tape will stop and you may place the next word on tape by pressing ENTER or type "E" and ENTER when you have finished.

Using The Program

Using the program is easy. Remember to remove the ear plug from the interface cable and simply follow the instructions. The program prompts you when to spell the word, congratulates you for a correct answer, and asks you to try again for an incorrect answer. Not included in the displayed instructions is the option to skip a word by entering "S" at the SPELL prompt. The program ends by telling you how many errors and skips were made.

SP.DO file is opened to allow you to monitor your child's (or your) progress. The file will contain the dates *Spelling Practice* was used, words skipped, and words misspelled in the format "correct spelling/incorrect spelling."

(Mel Perkins is an electrical engineer with Oklahoma Gas and Electric Co. He has been writing BASIC *programs since he got his Model 100 nearly a year ago.)*

The listing:

```
10 DATA "SPELLING","TEST","PROGRAM","FOR
","PCM","STOP"
20 CLS:PRINT@10,"SPELLING PRACTICE":PRIN
T@90,"LOAD SPELLING TAPE":PRINT@130,"REW
IND TO START":PRINT@170,"PRESS PLAY":PRI
NT@210,"PRESS ENTER TO START":PRINT@250,
;:INPUTDM$:CLS
25 PRINT@176,"Please stand by...":MOTORO
N:FORX=1TO3200:NEXTX:MOTOROFF:CLS
30 IF DM$="LOAD"THEN500
40 OPEN"RAM:SP.DO"FORAPPENDAS1
45 PRINT#1,"":PRINT#1,DATE$
50 READW$:IFW$="STOP"THENPRINTE;" ERRORS
":PRINTS;" SKIPS":CLOSE:END
60 MOTORON:FORX=1TO1800:NEXTX:MOTOROFF
65 LINE(66,32)-(167,32)
70 PRINT@132,;:LINEINPUT"SPELL ";AN$
75 IFAN$="S"THENPRINT#1,"SKIP-";W$:S=S+1
:CLS:GOTO100
80 IFAN$<>W$THENCLS:PRINT@212,"TRY AGAIN
":PRINT#1,W$;"/";AN$;" ";:E=E+1:GOTO65
90 CLS:PRINT@132,"GOOD!":FORX=1TO700:NEX
TX
100 PRINT@132,"PRESS ENTER TO CONTINUE":
PRINT@172,;:INPUTDM$:CLS:GOTO50
500 PRINT@40,;:LINEINPUT"PRESS ENTER TO
CONTINUE,E TO END";DM$:IFDM$="E"THENEND
510 MOTORON:FORX=1TO1800:NEXTX:MOTOROFF:
GOTO500
```

PCM

Planning Your Car's Maintenance Program (Or Doing What You AUTO.DO)

By Robert Frowenfeld
PCM Contributing Editor

For the last several months I've been giving you ideas and programs to help you get the most out of your Model 100 when you're out of the office and literally *on the road*. I've covered everything from economic analyses to grocery shopping, surveying to stocks. And this month I've come up with a little gem which will help *keep* you happily motoring along while you are (you guessed it) *on the road*. This month's program is designed to help you keep track of the maintenance on your automobile.

As usual, this program uses data files two data files to be exact. The first one, called AUTO.DO, identifies the various types of maintenance you're concerned about. Looking at my version of this file (Figure 1), I have listed 10 maintenance items that I would like to monitor. I entered these items with the *TEXT* program that comes with the Model 100. For every maintenance item, I entered the line number (1,2,3, etc.), a brief description in quotes, and two numbers. The first number is the interval in months, the second is the interval in miles for which the particular service needs to be performed. For example, line 6 indicates that the car should have its tires rotated every six months or 5,000 miles.

The second data file is called MAINT .DO: this data file is a transaction file which keeps track of all the maintenance done on my automobile. Included in each record is the date the service was performed, the number of the type of service (from AUTO.DO), an optional comment line, and the mileage (see Figure 2).

When you first run this program (I called my version *AUTO.BA*) you see a menu that offers five options (see Figure 3). The first option is for entering service that has been performed. Referring to Figure 4, you can see that the program requests the necessary information to construct a record for the maintenance file.

Every time you have work done to your car, you enter the appropriate date, comments (if any) and mileage. That way, when you invoke the second option of this program "Summarize Transactions," you can retrieve (and send to an attached printer) a complete, documented history of your car's maintenance record. As you can see from Figure 5, you can print out a very attractive service record or your car's maintenance records.

Option 3 from the main menu will list to the screen (or a printer) the various maintenance categories. Similar to Figure 1, this listing gives a nicely formatted view of the different types of maintenance and the respective time and mileage intervals at which they are to be performed (see Figure 6).

Now for you early birds who like to plan your car maintenance duties far into the future, I have included a special function that I think you may find useful. Option 4 from the main menu allows you to predict the different points at which your car will require maintenance. Referring to Figure 7, you simply type in the current month and year, along with the automobile's current

(Robert Frowenfeld owns his own computer programming firm in Louisville, Ky., and has completed his graduate course work in computer science at the University of Louisville.)

mileage. The program then asks if you want to predict the points in time or mileage at which the various types of maintenance will have to be performed. If you type in a number between 1 and 72, the program assumes you want to predict your maintenance schedule in months. However, if the number you enter is over 1000, then the program assumes you want the mileage readings at which to take care of your baby. Figure 8 is a partial listing of the recommended points in time (in months) that service is recommended over the next 36 months for my car. Note that the program indicates the last time each service was performed. As you may have guessed, the program compares the last time each particular type of service was performed and (based on the choice of months or mileage) uses the month and miles intervals entered into the file AUTO.DO to predict the next service period.

Well, there you have it. Here's hoping you'll continue to have happy motoring ahead of you . . . *on the road.*

Figure 1 — Auto.do file

```
1 "Oil Change",12,10000
2 "Oil Filter",12,10000
3 "Lubrication",12,10000
4 "Air Filter",6,5000
5 "Gas Filter",36,30000
6 "Rotate Tires",6,5000
7 "Inspect Brakes",12,10000
8 "Tune-up",24,20000
9 "Check Tire Pressure",6,5000
10 "Replace Shocks",36,30000
```

Figure 2 — Maint.do file

```
"1/1/83","1","10w40 - 5 quarts", 10000
"1/1/83","2","None", 10000
"3/15/83","8","Spark Plugs,etc.", 18000
"4/1/83","6","Took off snows", 20000
"6/10/83","4"," ", 24000
"6/10/83","5"," ", 24000
"8/1/83"," 1","5 quarts 10w40", 28000
"8/1/83"," 2"," ", 28000
```

Figure 3 — Menu

```
Automobile Maintenance Scheduler

  1  Enter Transaction
  2  Summarize Transactions
  3  List Categories
  4  Identify Maint. Requirements
  5  End Program
     Select:
```

Figure 4 — Transaction Entry

```
            Enter Transaction

Date       :8/1/30     Category #: 2
Cat. Desc. :Oil Filter
Comments   :
Mileage    :28000

        Entry Correct (Y/N):
```

Figure 5 — Partial Listing of Complete Maintenance History

```
      Account:  - Oil Change

  Date      Comments              Milage
-------- -------------------- --------
1/1/83   10w40 - 5 quarts       10000
8/1/83   5 quarts 10w40         28000

Total # entries =  2

      Account:  - Oil Filter

  Date      Comments              Milage
-------- -------------------- --------
1/1/83   None                   10000
8/1/83                          28000

Total # entries =  2

      Account:  - Air Filter

  Date      Comments              Milage
-------- -------------------- --------
6/10/83                         24000

Total # entries =  1
```

Figure 6 — Recommended Service Intervals

Maint#	Description	Mos.	Miles
1	Oil Change	12	10000
2	Oil Filter	12	10000
3	Lubrication	12	10000
4	Air Filter	6	5000
5	Gas Filter	36	30000
6	Rotate Tires	6	5000
7	Inspect Brakes	12	10000
8	Tune-up	24	20000
9	Check Tire Pres	6	5000
10	Replace Shocks	36	30000

Figure 7 — Input to Predict Maintenance Needs

```
      Identify Maint. Requirements

      Today's Date (mm/yy): 1/84

      Present Mileage    : 32000

      Predict (months or miles): 36
```

Figure 8 — Partial Listing of Predicted Maintenance Needs

```
Oil Change Schedule:  1/84  -  1/87

Last Maintenance was on:  8/83
Service should be every 12 months

Future Service Dates:  8/84  8/85  8/86

        Press any key to continue:
```

The listing:

```
1 MAXFILES=2:CLEAR 1000:DEFINT I-N:DEFST
R A,R,U
2 BL$=STRING$(38," ")
5 ES$=CHR$(27):R=ES$+"p":U=ES$+"q":GOTO
35
6 LINEINPUT IN$:X=VAL(IN$):IF IN$<>"" TH
EN Y=ASC(IN$):RETURN ELSE RETURN
35 '
40 FI$="maint.DO"
50 DATA"Enter Transaction","Summarize Tr
ansactions","List Categories","Identify
Maint. Requirements","End Program"
51 FOR I=1 TO 5:READ M0$(I):NEXT I
60 KEY 1," "+CHR$(13)
70 G1$="\      \ \                    \
 #####
71 H1$="  Date        Comments
Milage"
72 H2$="Maint# Description          Mos.
 Miles
73 G2$="  ##   \             \       ##
 #####"
74 H3$="------ ------------------------
-------"
75 H4$="-------- -------------------- -
-------"
100 CLS:PRINT@5,"Automobile Maintenance
Scheduler"
105 CLOSE
110 FOR I=1 TO 5:PRINT@45+I*40,R;I;U" "M
0$(I);:NEXT I
115 PRINT@289,"Select: ";
120 A=INPUT$(1):FX=VAL(A):IF FX<1 OR FX>
5 THEN 120
130 ON FX GOTO 1000,2000,3000,8000,4000
200 CLS:J=LEN(M0$(FX)):PRINTTAB(20-J/2);
M0$(FX):RETURN
300 OPEN FI$ FOR INPUT AS 1:RETURN
310 OPEN "auto.DO" FOR INPUT AS 2:RETURN
400 PRINT@290,"Press "R" F1 "U" to Exit"
;:RETURN
500 'pause
510 PRINT@280,"        ";R;" Press any ke
y to continue:";U;" ";:A=INPUT$(1):RETUR
N
600 'heading
610 CLS:PRINTH2$:PRINTH3$:IF (PR) THEN L
PRINT H2$:LPRINT H3$
620 RETURN
650 CLS:PRINT H1$:PRINT H4$:IF (PR) THEN
 LPRINT TAB(5)"Account: ";AC$;" - ";A3$:
LPRINT:LPRINT H1$:LPRINT H4$
660 RETURN
1000 'enter transaction
1005 CLOSE:OPEN FI$ FOR APPEND AS 1:GOSU
B 310
1010 GOSUB 200
1020 PRINT@080,"Date       :";TAB(22)"Ca
tegory #:";
1030 PRINT@120,"Cat. Desc. :";
1040 PRINT@160,"Comments   :";
1050 PRINT@200,"Mileage    :";
1060 GOSUB 400
1100 PRINT@92,"";:GOSUB 6:IF Y=32 THEN 1
00 ELSE IF LEN(IN$)<6 THEN 1100 ELSE DT$
=IN$:PRINT@280,BL$;
1110 PRINT@114,"";:GOSUB 6:IF IN$="" THE
N 1110 ELSE Y=ASC(IN$):IF Y=32 THEN 1110
1112 GOSUB 5000:IF ER=1 THEN 1110
1115 PRINT@132,A3$;:AC$=STR$(N)
1130 PRINT@172,"";:GOSUB 6:IF IN$="" THE
N 1130 ELSE DE$=IN$
1140 PRINT@212,"";:GOSUB 6:IF IN$="" THE
N 1140 ELSE AM!=X
1160 PRINT@288,"Entry Correct (Y/N): ";:
IN$=INPUT$(1):IF IN$="N" OR IN$="n" THEN
1000 ELSE IF IN$<>"Y" AND IN$<>"y" THEN
1160 ELSE PRINT IN$;
1170 PRINT#1,CHR$(34)DT$CHR$(34)","CHR$(
34)AC$CHR$(34)","CHR$(34)DE$CHR$(34)","A
M!
1180 GOTO 1000
2000 'summary
2010 CLOSE:GOSUB200:GOSUB300:GOSUB400:GO
SUB310:AA=""
2015 PRINT@206,"(Enter '*' for all accou
nts)"
2020 PRINT@130,"Enter Account #: ";:GOSU
B 6:IF IN$="" THEN 2020 ELSE IF Y=32 THE
N 100
2030 PRINT@280,BL$;
2035 IF IN$="*" THEN AA="*":GOTO 2045
2040 GOSUB 5000:IF ER=1 THEN 2000
2045 PRINT@200,BL$;:PRINT@129,"Send to p
rinter (Y/N): ";
2047 A=INPUT$(1):IF A=" " THEN 100 ELSE
IF A="N"OR A="n" THEN PR=0 ELSE IF A="Y"
 OR A="y" THEN PR=-1 ELSE 2045
2049 IF AA="*" THEN 6000
2050 GOSUB 650:LC=0:TS=0
2055 PRINT@280,"Acct.: "N" - "A3;:PRINT@
80,"";
2060 IF EOF(1) THEN 2092
2070 INPUT#1,DT$,DU$,DE$,AM!:N=VAL(DU$):
IF N<>XX THEN 2060
2080 PRINT USING G1$;DT$;LEFT$(DE$,21);A
M!:TS=TS+1:IF (PR) THEN LPRINT USING G1$
;DT$;LEFT$(DE$,21);AM!
2085 LC=LC+1:IF LC=5 AND (NOT PR) THEN G
```

```
OSUB 500:LC=0:GOSUB 650
2090 GOTO 2060
2092 CLOSE 1:IF AA="*" AND TS=0 THEN 209
8
2095 IF (NOT PR) THEN GOSUB 500:CLS:PRIN
T@5,"Acct. #";XX;" - ";A3$:PRINT@125,"To
tal # entries = ";TS:GOSUB 500
2097 IF (PR) THEN LPRINT:LPRINT"Total #
entries = ";TS
2098 IF AA="*" THEN IF (PR) THEN FOR I=1
TO5:LPRINT:NEXT I:GOTO 6000 ELSE 6000
2099 GOTO 2000
3000 'list accounts
3010 GOSUB 200:CLOSE:GOSUB 310:GOSUB 400
3020 PRINT@129,"Send to printer (Y/N): "
;
3030 A=INPUT$(1):IF A=" " THEN 100 ELSE
IF A="N"OR A="n" THEN PR=0 ELSE IF A="Y"
 OR A="y" THEN PR=-1 ELSE 3030
3035 PRINT A;:PRINT@280,BL$;
3040 GOSUB 600:LC=0
3050 IF EOF(2) THEN 3095
3060 INPUT#2,N,A1,M1,M2:PRINT USING G2$;
N,A1,M1,M2
3065 LC=LC+1:IF LC=5 AND (NOT PR) THEN G
OSUB 500: IF A=" " THEN 100 ELSE LC=0:GO
SUB 600
3070 IF (PR) THEN LPRINT USING G2$;N,A1,
M1,M2
3075 GOTO 3050
3095 GOSUB 500:CLOSE:GOTO 100
4000 'end
4010 CLS:MENU
5000 'search for category
5010 ER=0:IF EOF(2) THEN ER=1:RETURN
5020 INPUT #2,N,A3,M1,M2
5030 XX=X
5040 IF N=X THEN RETURN ELSE 5010
6000 ' for all accounts
6010 IF EOF(2) THEN 2000
6020 INPUT #2,N,A3,M1,M2
6025 XX=N:GOSUB7000:IF TS=0 THEN 6010
6030 CLOSE 1:GOSUB 300:ID$=A:GOTO 2050
7000 'check if any for this category
7010 TS=0:CLOSE 1:GOSUB 300
7015 IF EOF(1) THEN RETURN
7020 INPUT#1,DT$,DU$:IF VAL(DU$)=N THEN
TS=1:RETURN ELSE 7015
8000 'predict maintenance
8010 CLOSE:GOSUB200:GOSUB310:GOSUB400
8020 PRINT@80,BL$;:PRINT@86,"Today's Dat
e (mm/yy): ";:GOSUB6:IF IN$=" "THEN RUN
ELSE J=INSTR(IN$,"/"): IF J<2 OR J>3 THE
N 8020
8030 MM=X:YY=VAL(MID$(IN$,J+1)):IF MM<1
OR MM>12 OR YY<60 OR YY>99 THEN 8020
8035 PRINT @160,BL$;:PRINT @166,"Present
 Mileage     : ";:GOSUB 6:IF IN$=" " THE
N RUN ELSE IF X<=0 OR X>500000 THEN 8035
 ELSE ML=X:CM=X
8040 PRINT@280,BL$;:PRINT@240,BL$;:PRINT
@246,"Predict (months or miles): ";:GOSU
B 6
8050 P=X:IF (P>=1 AND P<=72) OR (P>=1000
 AND P<=50000) THEN 8055 ELSE 8040
8055 IF P<100 THEN OP=1 ELSE OP=2
8060 CLS
8070 CLS:IF EOF(2) THEN RUN
8080 INPUT #2,N,A1,M1,M2
8090 GOSUB 9000
8095 ON OP GOTO 8100,8200
8100 'predict for months
8110 IF LD+M1>YY*12+MM+P THEN 8070
8120 PRINT A1;" Schedule: ";:PD=YY*12+MM
:GOSUB8300:PRINT" - ";:PD=PD+P:GOSUB8300
:PRINT
8130 PRINT:PRINT "Last Maintenance was o
n: ";:IF LD=0 AND LM=0 THEN PRINT"----":
GOTO 8150
8140 PD=LD:GOSUB 8300:PRINT
8150 PRINT"Service should be every"M1"mo
nths":PRINT
8160 PRINT"Future Service Dates: ";:DD=L
D:IF LD+M1<YY*12+MM THEN DD=YY*12+MM-M1:
GOTO 8180
8170 IF DD+M1>LD+P THEN 8190
8180 DD=DD+M1:PD=DD:GOSUB 8300:GOTO 8170

8190 GOSUB 500:GOTO 8070
8200 'predict for miles
8210 IF LM+M2>CM+P THEN 8070
8220 PRINT A1;" Schedule: ";CM;"-";CM+P
8230 PRINT:PRINT "Last Maintenance was a
t: ";:IF LD=0 AND LM=0 THEN PRINT"----":
GOTO 8250
8240 PRINT LM;"miles"
8250 PRINT"Service should be every"M2"mi
les":PRINT
8260 PRINT"Future Service @ (Miles): ";:
DD=LM:IF LM+M2<CM THEN DD=CM-M2:GOTO 828
0
8270 IF DD+M2>LM+P THEN 8290
8280 DD=DD+M2:PRINT USING" ##### ";DD;:G
OTO 8270
8290 GOSUB 500:GOTO 8070
8300 'print date
8310 J=PD MOD 12:K=INT(PD/12):IF J=0 THE
N J=12: K=K-1
8320 PRINT USING"##/## ";J;K;:RETURN
8999 GOTO8999
9000 'find last maint date/mileage
9005 CLOSE 1:GOSUB 300
9010 LD=0:LM=0
9020 IF EOF(1) THEN RETURN
9030 INPUT #1,DT$,DU$,DE$,AM!:IF VAL(DU$
)<>N THEN 9020
9040 LM=AM!:LD=VAL(DT$)+VAL(RIGHT$(DT$,2
))*12
9050 GOTO 9020
```

PCM

Portable Mass Storage Is Here

By Jim Hawk

If you got together a room full of Model 100 users and asked them what their main need is, 90 percent would probably reply "Mass Storage!"

Virtually all of us have encountered difficulties using cassette. Either it's hard to locate something, it won't load, or takes too long — cassette is just not the best means for mass storage.

Not too much can be done about the 32K maximum RAM of usable onboard memory, although the M100 look-alike, the NEC 8201A, does have an option for plug-in 32K RAMs with battery backup. The catch is cost: $400 per 32K when the same amount of data can easily be put on a two dollar cassette. So, it boils down to finding some other means of magnetic storage for a cost effective way to improve on the cassette. Enter the wafer tape. . . .

Wafer Promises Disk-Like Speed, Cassette Size And Cost

A wafer drive for the Model 100 has already been announced but wafer drives have been around since Radio Shack's original Model 1. Several thousand reportedly were sold by a third-party company called Exatron, and also connected to original Apples before disk drives were common. Wafer tape drives utilize the continuous loop format made popular by 8-track audio tapes. About the size of a business card with tape as thin as that used on a microcassette, wafer tapes use only one spool with around 20 feet of tape connected back onto itself by a strip of metal foil. This "cartridge" is inserted into a tape drive unit that normally runs at a speed of around 10 inches per second, but can rev up to 20 inches per second in the fast forward/search mode (recueing at the foil strip). This speed, combined with a digital method of encoding data, makes for a combination that runs circles around cassettes and gives disk drives (especially portable ones) a run for the money.

Larry Holmes, founder of Holmes Engineering out of Salt Lake City now marketing a wafer drive for the Model 100, says the need for the product became evident just from general observation. And he decided to develop the wafer drive concept after checking out portable disk drive units and conclud-

---

(Jim Hawk has been working in radio news for tha past 12 years and has a science and electronics background. He also does freelance writing in Washington, D.C.)

ing the cost would end up being $700 to $1,000 . . . a lot of money for any M100 peripheral. The Holmes wafer drive, selling for just under $350, is actually almost a computer itself. Explains Holmes, "we saw some quirks in the Model 100 that didn't lend themselves to easy solution, so it became obvious it would have to be fairly complex . . . fairly smart." The expansion connector on the bottom of the 100 would have simplified things, but Holmes decided it just wasn't usable for day-to-day applications. So, the RS-232 port on the back was the only other way to connect the two machines.

Holmes also knew it meant he'd have to have a microprocessor in his drive, because the 100 doesn't have any "hardware handshaking" on the RS-232 port — which means you don't have any way for the two machines to talk back and forth to each other and let each know that data has been properly received. What Holmes ended up with is a box about two-thirds the size of the 100, containing the wafer drive, a Z-80 based microprocessor, 16K of RAM, a separate ROM and a rechargeable battery.

If the unit lives up to specifications, it will not only quickly store and retrieve text and programs, it'll give you a directory of what files are on the tape as well as recheck each and every byte before completing a cycle. This will theoretically make the wafer drive operate very much like a disk system, but critics won't be silenced until they see some in-field proof. Wafer drives have been around a long time, but have only found acceptance in the engineering and programming fields. Also, no standard exists: Hewlett Packard currently markets a mini-digital cassette drive of its own, but it uses a two-spooled minicassette that can shuttle back and forth instead of just in a single continuous direction and naturally connects only to HP products. Holmes says a standard-sized wafer can hold about 20K of information, and that loading runs at 9600 Baud . . . over six times the 1500 Baud speed of the cassette port. Part of this speed is made possible by the 16K RAM, which utilizes about 12K as a "buffer," allowing the Holmes drive to take in data faster than it's storing it on tape. Will the Holmes wafer drive be able to compete with upcoming disk systems? One bad omen comes from the ill-fated Texas Instruments consumer line: just before they dropped their low-end machines, TI was advertising a wafer-based digital tape drive for its notebook-sized CC-40 (still available, but the drive was dropped before it reached the stores). Despite that debacle, inventor Holmes is optimistic — saying he took a propotype unit to the Comdex computer show recently and found many interested parties. Also the maker of the basic wafer drives (now under the name of Entrepo in Sunnyvale, Calif.) has indicated it intends to go to Hong Kong for large scale manufacturing — and that may bring further price reductions.

Don't Count Out The Disk

At that same Comdex convention in the fall — one of the major events in the computer industry — NEC was showing off a prototype that drew a lot of attention: a portable, battery-powered disk drive. Although not available till the summer, the specifications on the unit are impressive: using 3½-inch double-density dual-sided disks, the machine promises 160K of storage per side, meaning a total of 320K per disk. The PC-8031 connects to the company's Model 100 brother (the 8201A) using a port not appearing on the 100: an 8-pin version of the modular jack you see on telephones . . . meaning incompatibility. But I have word from a source wishing not to be disclosed that either a third-party firm or Tandy Corporation is expected to offer a similar attachment for the Model 100 — exactly when is not known. What we do know is that some big Japanese firms are struggling right now over a standard for the so-called "microfloppy" disk.

> ". . . either a third-party firm or Tandy Corporation is expected to offer a . . . [disk drive] attachment for the Model 100 — exactly when is not known."

The Incredible Shrinking Disk

"Floppy" disks — thin sheets of mylar coated with metal oxide and encased in a paper envelope — first were employed about 10 years ago to load software into

large mainframe computers.

These 8-inch diameter monsters operated in the clean-room conditions maintained for the mainframes, using trained operators. It was only later that floppies went through their first shrinking act, to 5¼ inches diameter, to be adapted for small personal computers. But the current machines suffer from the lack of dust-free environments, the fragility of the mylar disks, and relatively complicated operating procedures. Just about any computer repairman will tell you the disk drive is the first thing to require maintenance. And how many times have you heard from personal computer users that you should always make backup copies of disks because of their vulnerability? The Japanese have done some clever things to the new generation of floppies: of course, shrinking them in size while increasing capacity, but also encasing the microfloppies inside rigid plastic shells. Myles Tintle, general manager of Sony Corporation's communication products division in Park Ridge, N.J., compares the development to audio when cassettes took over the open reel format. These new floppies or "diskettes" are not only protected from oily fingers and dust, their rigid plastic cases are also designed so they can't be loaded upside down. And the drives (like 300 RPM turntables) no longer use belts and pulleys — they instead employ direct-drive brushless d.c. motors.

Also, instead of grabbing onto the mylar itself to spin the disk past a fixed head, a metal hub has been employed to take the stress away from the recording medium. Combined with a system requiring only six watts maximum, three watts standby, Sony has made a portable possible . . . although Tintle admits a battery-powered unit has yet to be unveiled. Already, Hewlett Packard has put Sony's drives on its new HP-150 as well as three other HP products. Sony's Tintle says his firm has now shipped in excess of 150,000 units, and they expect to ship another 500,000 in '84. Competing against Sony's 3.5-inch microfloppy, Hitachi has a 3-inch version aimed directly at the portable computer market. The marketplace will probably have a clearer choice as the year progresses, but ANSI (the American National Standards Institute) has started work on drafting a 3.5-inch standard. We already own Japanese computers — the Model 100 and the NEC 8201A are both made by Kyocera — and it seems a miniature Japanese disk machine may be on the horizon as well. But if you own a desktop personal computer like Apple or IBM, you can get your 100 talking to those disk units right now. . . .

DISK+ From The *WRITE+* People

Software reviewers are in unanimous agreement: the *WRITE+* software from the Portable Computer Support Group in Dallas makes the 100 a complete word processor. Now, the same company has just announced *DISK+* for the IBM PC, TRS-80 Model 4, Apple IIe and others. This software, which arrived for review the day this article had to go to print, promises to transfer Model 100 files "with merely a function key." You can, with the aid of an RS-232 cable with "null modem," transfer any file from RAM to disk, or vice versa. PCSG's Sam Redman told me "our *DISK+* gives you guaranteed data transfer because it checks every line until correct . . . at 19000 Baud, almost instantly." Already being used by staffers, Redman says he has a rule: nobody uses cassette anymore, unless it's to load *DISK+*. More on that software development in a later issue.

A Final Word On The Lowly Cassette

Wouldn't it be nice to be able to cue your cassette directly to the program or text file desired? This author has been speeding up cassette operation by a simple trick: voice markers at the beginning and end of files. The only requirement is a cassette recorder with a built-in mike: I stumbled upon this because the Radio Shack CTR-60 was on sale for $30 when I bought my 100, and I couldn't see any mechanical difference between it and the twice-as-expensive CCR-81 which is specifically for computers but doesn't have the built-in mike. More out of curiosity than anything, I tried talking about the file I was about to put on tape, left a blank space, then saved my file normally. To my surprise, the M100 loaded those files about 80 percent of the time, even when I didn't cue up past the voice-announce. Now it's become so routine, I easily jump past huge text files and find exactly what I want on tape . . . no waiting for "skip:XXX." It's easier done than said — first you need to become familiar with exactly what a BASIC or text file sounds like. Listening at normal speed, computer files sound like jamming signals on shortwave radio. But at fast forward, the sound is more like hissing air. Anyway, text files have a characteristic on/off sound, while BASIC files have a short identifier blip at the beginning and then a continuous sound till finished. (I was reassured by the people at Holmes Engineering that fast-forwarding does *not* deteriorate the magnetic integrity of a file: both their wafer drive and other disk drives employ heads that stay in contact with the medium.) The technique: get ready to save a file in the normal fashion, including putting the machine on "Record." Then, simply unplug the "remote" cable and let it roll in silence for a moment, then also unplug the "Aux" connector, automatically switching open the built-in mike. Say whatever you want about the file: its six-character name, the date and maybe even the size of the file. Then, reverse the plug-in order, first re-connecting the Aux plug, and letting it roll a few moments in silence before also re-connecting the remote plug. SAVE in the usual fashion, and if you want, put a voice billboard on the end of the file as well. I've found this procedure to be extremely helpful, since I can cue up right to the point desired, and if it doesn't load right away I know to go back and try again at a slightly different volume level. Riding *past* the voice-announce also improves loadability, to near 100 percent. To listen back, unplug remote again and also unplug the black earphone plug. All this requires a routine and some manual dexterity, but once you get the hang of it I don't think you'll go back to the unmarked system. After all, the ultimate "personal computer" is attached right to your shoulders, complete with total voice recognition — why not make good use of it?

Future Mega-Storage

While the longest Holmes wafer tape can hold up to 70K, and the double-sided Sony microfloppy holds an even megabyte, there's no such thing as enough capacity. Sony's plan for the next step is to double storage capacity to two megabytes per 3.5-inch disk, and beyond that, something called "vertical recording" promises five to six megabytes per disk . . . similar to some hard disk Winchester systems. 1984 promises to be a big year for portable computing and some remarkable peripherals: stay tuned. PCM

FOR FAST AND RELIABLE DATA ENTRY
GET RADIO SHACK'S NEW BAR CODE

Our TRS-80 Model 100 is one of the most talked-about, most-wanted portable computers ever! The reason is simple.

We started with a truly portable workstation with five programs built in. A four lb. "wonder" that runs on batteries or optional AC adapter, that's small enough to fit in your briefcase with room to spare.

Now we've added still another exciting dimension to the TRS-80 Model 100.

With the new Model 100 Bar Code Reader, your data input tasks are made easier, faster and more concise than with conventional keyboard data entry.

This optical scanning device is ideal for certain retail, wholesale, manufacturing and administrative applications.

EADER FOR THE TRS-80® MODEL 100

Billing, couponing and inventory control are made more efficient for warehouses and supermarkets.

And the tracking and tracing of manufactured goods becomes far easier and more cost-effective.

Even libraries can use the Model 100 Bar Code Reader for document storage, retrieval and tracking.

The pen-like bar code wand is easy to use. Plug it into Model 100's standard bar code reader interface. Load the included cassette software drivers (recorder required) – that are callable from Model 100's built-in BASIC. And then begin scanning bar codes for speedy input.

Read Universal Product Code, 3 of 9, or Plessey bar code formats that are either lithographed or computer-printed.

And the best part of using the Model 100 Bar Code Reader is the Model 100.

With the built-in BASIC, you can create programs to analyze bar code data.

You can then prepare reports with Model 100's personal word processing program.

With the built-in modem, you can easily transmit your latest bar code files and reports to any host computer.

In addition, Model 100 is an address/phone directory, appointment scheduler, and telephone auto-dialer.

Stop by your nearest Radio Shack Computer Center, participating store or dealer and learn more about all the options a Model 100 Bar Code Reader will give you.

Model 100
Bar Code Reader

99.95
26-1183

Send me a free TRS-80 Catalog today!

Mail To: Radio Shack, Dept. 84-A-730
300 One Tandy Center, Fort Worth, Texas 76102

NAME _____

COMPANY _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

TELEPHONE _____

8K Model 100
799.00
26-3801

24K Model 100
999.00
26-3802

Radio Shack®

The biggest name in little computers®
A DIVISION OF TANDY CORPORATION

Prices apply at participating Radio Shack stores and dealers.

SOLO — SOLITAIRE WITH SPICE

By Robert Delbourgo

How often have you waited around at airports because of a delay in your flight times? You've read through the papers, leafed through the magazines and still find you have time to kill — somehow.

Fortunately, you are carrying your cherished PoCo which contains all your work files, schedules, etc. PoCo also contains all you need to while away those extra minutes by carrying programs stored in it's memory! Here is a program to help you pass the time. . . .

It is based upon the traditional solitaire game, but I have spiced it up to make every game different and interesting. Otherwise (once you have cracked the method, that is) solitaire tends to become a boring ritual. *Solo*, as I've called my offering, will, I believe, challenge you with its variety of settings.

Firstly, you have the option of choosing between six different board layouts. Secondly, there are five wild pieces, randomly sprinkled at the start of every game; these are worth five points each, in contrast to the normal one point pieces. The aim of the game is to eliminate all pieces but one — preferably with the last piece being a normal one and remaining at the center of the board. If you succeed in doing so, your score will reduce to zero.

The program is self-prompting, but I should remind you that to remove a piece you must place your cursor (using arrow keys) over the blank spot into which the jumping piece will fall. When you press R or L or U or D to displace the jumper, do so *very deliberately*, the reason being that INKEY$ must be appropriately sampled.

I won't say a lot about the listing except to point out the subroutine 1200 which moves the cursor, subroutine 1250 which chooses the jumping piece and the subsequent subroutines which readjust the locations of the pieces. I have used the array A(I,J) to mark (by 0 or 1) whether a spot is occupied or not. When a move is legal a BEEP is heard. If you think this is likely to disturb your traveling companions, then delete BEEP from lines 1264, 1269, 1274 and 1279. The program makes extensive use of the LINE functions, another triumph for Microsoft ingenuity.

(Robert Delbourgo is a professor of physics and computer advocate from Hobart, Tasmania, Australia. His son, Tino, is the author of a program which appears elsewhere in this issue.)

```
10 FORI=1TOINT(VAL(RIGHT$(TIME$,2))):R=R
ND(1):NEXTI:DIMA(9,9)
20 CLS:GOSUB1000
30 S$="SSSSS":O$="OOOOO":L$="LLLLL"
40 FORI=42TO202STEP40:PRINT@I,S$;:PRINT@
I+6,O$;:PRINT@I+12,L$;:PRINT@I+18,O$;:NE
XTI
50 PRINT@83,SPACE$(4);:PRINT@162,SPACE$(
4);:FORI=89TO169STEP40:PRINT@I,SPACE$(3)
;:PRINT@I+12,SPACE$(3);:NEXTI:FORI=55TO1
75STEP40:PRINT@I,SPACE$(4);:NEXTI
60 PRINT@282,"Variations on Solitaire";
70 SOUND2384,20:SOUND2384,20:SOUND2488,2
0:SOUND3134,40:SOUND3134,40:SOUND2488,20
:SOUND2488,20:SOUND2793,20:SOUND3516,40:
SOUND3516,40
80 GOSUB1050:PRINT@0,"SOLO-Bob Delbourgo
, Hobart,";:PRINT@45,"Tasmania,AUSTRALIA
7005";:PRINT@80,STRING$(28,167);
90 PRINT@120,"Reduce your score to 0, to
 ";:PRINT@160,"end with 1 piece at centr
e.";:PRINT@200,"d counts 1, o counts 5 a
nd";:PRINT@240,"there are 5 wild o's at
the";:PRINT@280,"start of each game. REA
DY?..";:FORT=1TO3000:NEXT
100 GOSUB1050:PRINT@0,"    SOLO - BOARD
LAYOUT";:PRINT@40,STRING$(28,167);:PRINT
@280,"<ESC> to end game anytime.";
110 PRINT@80,"1.Full board  2.Diamond";:
PRINT@120,"3.Vertical +  4.Diagonal X";:
PRINT@160,"5.O-shaped    6.Swastika";:PR
INT@240,STRING$(28,167);
120 PRINT@200,"ENTER YOUR CHOICE";:INPUT
C:C=INT(C):IFC<1ORC>6THEN120
130 ONC GOTO150,200,250,300,350,400
150 N=99+N:GOTO600
200 N=59+N:FORI=1TO4:FORJ=0TO8STEP8:LINE
(7*I+165,7*J+2)-(7*I+168,7*J+5),0,BF:LIN
E(7*I+200,7*J+2)-(7*I+203,7*J+5),0,BF:A(
I,1+J)=0:A(I+5,1+J)=0:NEXTJ,I
210 FORI=1TO3:FORJ=2TO8STEP6:LINE(7*I+16
5,7*J-5)-(7*I+168,7*J-2),0,BF:LINE(7*I+2
07,7*J-5)-(7*I+210,7*J-2),0,BF:A(I,J)=0:
A(I+6,J)=0:NEXTJ,I
220 FORI=1TO2:FORJ=3TO7STEP4:LINE(7*I+16
5,7*J-5)-(7*I+168,7*J-2),0,BF:LINE(7*I+2
14,7*J-5)-(7*I+217,7*J-2),0,BF:A(I,J)=0:
A(I+7,J)=0:NEXTJ,I
230 LINE(172,23)-(175,26),0,BF:LINE(228,
23)-(231,26),0,BF:A(1,4)=0:A(9,4)=0:LINE
(172,37)-(175,40),0,BF:LINE(228,37)-(231
,40),0,BF:A(1,6)=0:A(9,6)=0
240 GOTO600
250 N=63+N:FORI=1TO3:FORJ=1TO3:LINE(7*I+
165,7*J-5)-(7*I+168,7*J-2),0,BF:LINE(7*I
+207,7*J-5)-(7*I+210,7*J-2),0,BF:LINE(7*
I+165,7*J+37)-(7*I+168,7*J+40),0,BF:LINE
(7*I+207,7*J+37)-(7*I+210,7*J+40),0,BF
260 A(I,J)=0:A(I+6,J)=0:A(I,J+6)=0:A(I+6
,J+6)=0:NEXTJ,I:GOTO600
```

```
300 N=63+N:FORI=3TO7:LINE(7*I+165,2)-(7*
I+168,5),0,BF:LINE(7*I+165,58)-(7*I+168,
61),0,BF:LINE(172,7*I-5)-(175,7*I-2),0,B
F:LINE(228,7*I-5)-(231,7*I-2),0,BF:A(I,1
)=0:A(I,9)=0:A(1,I)=0:A(9,I)=0:NEXTI
310 FORI=4TO6:LINE(7*I+165,9)-(7*I+168,1
2),0,BF:LINE(7*I+165,51)-(7*I+168,54),0,
BF:LINE(179,7*I-5)-(182,7*I-2),0,BF:LINE
(221,7*I-5)-(224,7*I-2),0,BF:A(I,2)=0:A(
I,8)=0:A(2,I)=0:A(8,I)=0:NEXTI
320 LINE(200,16)-(203,19),0,BF:LINE(200,
44)-(203,47),0,BF:LINE(186,30)-(189,33),
0,BF:LINE(214,30)-(217,33),0,BF:A(3,5)=0
:A(5,3)=0:A(5,7)=0:A(7,5)=0
330 GOTO600
350 N=75+N:FORI=3TO7:FORJ=3TO7:LINE(7*I+
165,7*J-5)-(7*I+168,7*J-2),0,BF:A(I,J)=0
:NEXTJ,I
360 GOTO600
400 N=75+N:FORI=3TO4:FORJ=1TO3:LINE(7*I+
165,7*J-5)-(7*I+168,7*J-2),0,BF:LINE(7*I
+186,7*J+37)-(7*I+189,7*J+40),0,BF:A(I,J
)=0:A(I+3,J+6)=0:NEXTJ,I
410 FORI=7TO9:FORJ=3TO4:LINE(7*I+165,7*J
-5)-(7*I+168,7*J-2),0,BF:LINE(7*I+123,7*
J+16)-(7*I+127,7*J+19),0,BF:A(I,J)=0:A(I
-6,J+3)=0:NEXTJ,I
420 GOTO600
500 I=5:J=5:G=G+1:GOSUB1050:PRINT@0,"
SOLO - MOVEMENT RULES";:PRINT@40,STRING$
(28,167);:PRINT@80,"Use arrow keys to po
sition";:PRINT@120,"cursor over BLANK sp
ot, then";:PRINT@160,"press <U>,<D>,<R>
or <L> if";
510 PRINT@200,"the jumping piece is two"
;:PRINT@240,"spaces up, down, right or";
:PRINT@280,"left of the BLANK piece.";
515 I=5:J=5:FORT=1TO1000:NEXTT:PRINT@0,"
GAME #:"G;
520 IFN=0THEN550
530 GOSUB1150:GOSUB1200:GOSUB1250:GOSUB1
300
540 PRINT@10,"     SCORE :"N;:GOTO520
550 T=500:FORI=1TO5:SOUNDT,20:T=T*2:NEXT
I:FORI=80TO280STEP40:PRINT@I,SPACE$(28);
:NEXTI:PRINT@122,"Another round <Y/N>?";
555 I$=INKEY$
556 IFI$="Y"ORI$="y"THEN560
557 IFI$="N"ORI$="n"THENCLS:PRINT@40,"By
e bye then ..":END
558 GOTO555
560 CLS:GOSUB1000:GOTO100
600 FORK=1TO5
610 I=INT(9*RND(1)+1):J=INT(9*RND(1)+1)
620 IFA(I,J)=1THENLINE(7*I+166,7*J-4)-(7
*I+167,7*J-3),0,B:A(I,J)=5 ELSE610
630 NEXTK:GOTO500
1000 FORI=0TO63STEP7:LINE(170,I)-(233,I)
:LINE(170+I,0)-(170+I,63):NEXTI
1010 FORJ=2TO58STEP7:FORI=172TO228STEP7:
LINE(I,J)-(I+3,J+3),1,BF:NEXTI,J
1020 LINE(200,30)-(203,33),0,BF:FORI=1TO
9:FORJ=1TO9:A(I,J)=1:NEXTJ,I:A(5,5)=0:RE
TURN
1050 FORI=0TO280STEP40:PRINT@I,SPACE$(28
);:NEXTI:RETURN
1150 X=7*I+164:Y=7*J-6:LINE(X,Y)-(X+5,Y+
5),0,B:RETURN
1200 I$=INKEY$
1205 IFI$=CHR$(30)THENJ=J-1:IFJ<1THENJ=1
1210 IFI$=CHR$(31)THENJ=J+1:IFJ>9THENJ=9
1215 IFI$=CHR$(28)THENI=I+1:IFI>9THENI=9
1220 IFI$=CHR$(29)THENI=I-1:IFI<1THENI=1
1225 IFI$=CHR$(27)THEN550
1230 RETURN
1250 X=7*I+165:Y=7*J-5:T$=INKEY$
1251 IFT$="U"ORT$="u"THENGOSUB1261:RETUR
N
1252 IFT$="D"ORT$="d"THENGOSUB1266:RETUR
N
1253 IFT$="R"ORT$="r"THENGOSUB1271:RETUR
N
1254 IFT$="L"ORT$="l"THENGOSUB1276:RETUR
N
1255 RETURN
1261 IFJ<2THENRETURN
1262 IFA(I,J-1)=0ORA(I,J-2)=0ORA(I,J)<>0
THENRETURN
1263 LINE(X,Y-14)-(X+3,Y-11),0,BF:LINE(X
,Y-7)-(X+3,Y-4),0,BF:N=N-A(I,J-1):A(I,J)
=A(I,J-2):A(I,J-2)=0:A(I,J-1)=0
1264 BEEP:GOSUB1350:RETURN
1266 IFJ>7THENRETURN
1267 IFA(I,J+1)=0ORA(I,J+2)=0ORA(I,J)<>0
THENRETURN
1268 LINE(X,Y+14)-(X+3,Y+17),0,BF:LINE(X
,Y+7)-(X+3,Y+10),0,BF:N=N-A(I,J+1)
1269 A(I,J)=A(I,J+2):A(I,J+2)=0:A(I,J+1)
=0:BEEP:GOSUB1350:RETURN
1271 IFI>7THENRETURN
1272 IFA(I+1,J)=0ORA(I+2,J)=0ORA(I,J)<>0
THENRETURN
1273 LINE(X+14,Y)-(X+17,Y+3),0,BF:LINE(X
+7,Y)-(X+10,Y+3),0,BF:N=N-A(I+1,J)
1274 A(I,J)=A(I+2,J):A(I+2,J)=0:A(I+1,J)
=0:BEEP:GOSUB1350:RETURN
1276 IFI<2THENRETURN
1277 IFA(I-2,J)=0ORA(I-1,J)=0ORA(I,J)<>0
THENRETURN
1278 LINE(X-14,Y)-(X-11,Y+3),0,BF:LINE(X
-7,Y)-(X-4,Y+3),0,BF:N=N-A(I-1,J)
1279 A(I,J)=A(I-2,J):A(I-2,J)=0:A(I-1,J)
=0:BEEP:GOSUB1350:RETURN
1300 X=7*I+164:Y=7*J-6:LINE(X,Y)-(X+5,Y+
5),1,B:RETURN
1350 IFA(I,J)=1THENLINE(X,Y)-(X+3,Y+3),1
,BF
1360 IFA(I,J)=5THENLINE(X,Y)-(X+3,Y+3),1
,B
1370 RETURN
```

PCM

SIX FINANCIAL PROGRAMS FOR POCO

By William T. Stauffer

Do you have a dire need to buy new computer equipment or a new house with a larger room to work on your computer? Or do you have other needs requiring loan assistance from your "friendly" financial institution? Here is a group of financial programs to help you decide if you can, or almost can, afford it!

The program is divided into six programs as follows:

Program 1 — Payment Amount, computes the Payment Amount when Loan Value, Annual Percentage Rate (APR), Number of Payments Per Annum, and Length of Loan (in years) are given.

Program 2 — Present Value, computes Loan Value when APR, Payment Amount, Number of Payments Per Annum, and Length of Loan are given.

Program 3 — Number of Payments, computes Number of Payments when APR, Payment Amount, Number of Payments Per Annum, and Loan Value are given.

Program 4 — Amortization Schedule, lists on the display an abbreviated amortization schedule which includes Period (payment number), Principal (amount of payment applied to principal), Interest (amount of payment applied to interest), and Balance.

Program 5 — Printout, provides an amortization schedule printout. A header is printed on each sheet, showing the Loan Amount, Payment, APR, and Payments per annum. The following is printed in columnar format: Payment Number, Interest Payment, Principal Payment, Balance, Cumulative Interest, and Cumulative Principal. The program paginates the listing — each sheet listing 48 periods. For example, see Table 1.

(William T. Stauffer is an electromagnetic radiation effects projects engineer at the White Sands Missile Range in New Mexico. He has had more than 20 years experience working with computers — eight months with the Model 100 — and has published a number of computer programs and related articles, including one in the October issue of PCM.)

AMORTIZATION SCHEDULE

LOAN AMOUNT $1,200.00 — 16 % INTEREST
PAYMENT $108.88 — 12 PAYMENTS PER YEAR

PAYMENT NUMBER	INTEREST PAYMENT	PRINCIPAL PAYMENT	BALANCE	CUMULATIVE INTEREST	CUMULATIVE PRINCIPAL
0	$0.00	$0.00	$1,200.00	$0.00	$0.00
1	$16.00	$92.88	$1,107.12	$16.00	$92.88
2	$14.76	$94.12	$1,013.00	$30.76	$187.00
3	$13.51	$95.37	$917.63	$44.27	$282.37
4	$12.24	$96.64	$820.99	$56.51	$379.01
5	$10.95	$97.93	$723.06	$67.46	$476.94
6	$9.64	$99.24	$623.82	$77.10	$576.18
7	$8.32	$100.56	$523.26	$85.42	$676.74
8	$6.98	$101.90	$421.36	$92.40	$778.64
9	$5.62	$103.26	$318.10	$98.02	$881.90
10	$4.24	$104.64	$213.46	$102.26	$986.54
11	$2.85	$106.03	$107.43	$105.11	$1,092.57
12	$1.43	$107.43	$0.00	$106.54	$1,200.00

Program 6 — Credit Card Approximation, is an "orphan" program which approximates the cost and time of payoff for a credit card when balance and APR are given. This program assumes a fixed interest rate, payoff based on 18 months, one payment per month, minimum payment made each month, and minimum payment of $10.

These programs are all self-prompting and all that is required is to follow the instructions and enter data as requested. Upon the completion of each program, if a key number 1 through 5 is pressed (NOTE: these are number keys, NOT function keys), the program will automatically transfer to the selected program, if any other key is pressed, the program menu will be displayed. To end the program, press BREAK and then function key F8.

Programs 1 through 5 are interactive. When data is once entered, these data will remain in the computer and, when prompted for further data, if ENTER is pressed, the data as shown on the display will not be altered.

In programs 1 through 4, when the last data is entered, the Loan Value, Interest Rate, Payment Amount, Total Interest, Total Payments, and Number of Payments will be displayed. A similar display will be presented upon completion of the Amortization Schedule, Program 6. If a printer is connected, this display may be printed by pressing PRINT.

The program includes extensive use of spaces and tabs to provide a 40-column listing. This feature provides a more easily readable format on the PoCo display. To provide a "neat" printout, save the program in ASCII format and print it out from the .DO file as follows:

1) Load the FINANCIAL PROGRAMS program
2) Press SHIFT and BREAK
3) Press F3
4) Type: FINAN",A
5) Press F8
6) Press SHIFT and PRINT
7) Type: 40
8) Press ENTER

The complete program, shown in Program Listing 1, occupies 4738 bytes of memory, which may be a little rich for your blood if you have an 8K memory PoCo. The program may be modified to delete programs 5 and 6 and remove some of the "garbage" using the following procedure:

1) Delete lines 120-190.
2) Delete "FOR PRINTOUT PRESS 5 FOR AMORTIZATION SCHEDULE PRESS 6" from line 210.
3) Change "6" to "4" in line 240. Line 240 will then read:
   240 IF S<OORS>4 GOTO220
4) Delete lines 810-1510.

This truncated program occupies 1990 bytes of memory and is shown in Program Listing 2.

These programs were developed and used on a TRS-80, Model 1 several years ago and have been adapted for the use on the PoCo. They have been used many times as a "service" to friends and have been distributed (in the Model 1 format) to an appreciable number of other computer users.

If you would like to have tapes of the programs, forward a tape with a self-addressed, stamped return label to:

Bill Stauffer
204 Tooele
WSMR, NM 88002

If you want any additional information or to make any comments, call (505) 678-3966 or leave in CompuServe 70435,237.

Listing 1:

```
100 '*********************************
110 '*      FINANCIAL PROGRAMS       *
    *********************************
120 '*  EQUIPMENT USED:              *
130 '*      TRS-80 COMPUTER MODEL 100  *
140 '*      EPSON MX-80 PRINTER      *
    *********************************
150 '*      BY:                      *
160 '*      WM. T. STAUFFER          *
170 '*      204 TOOELE               *
180 '*      WHITE SANDS MR, NM 88002   *
190 '*      505 678-3966             *
    *********************************
200 CLS:
    PRINT@13,"MORTGAGE LOAN":
    A$="$$######.##"
210 PRINT"FOR PAYMENT AMOUNT
  PRESS 1 FOR PRESENT VALUE
  PRESS 2 FOR NUMBER OF PAYMENTS
  PRESS 3 FOR AMORTIZATION SCHEDULE
  PRESS 4 FOR PRINTOUT
PRESS 5 FOR CREDITCARD APPROXIMATION
PRESS 6
220 S$=INKEY$: IF S$="" GOTO220
230 S=VAL (S$)
240 IFS<1ORS>6 GOTO200
250 ONSGOTO260,300,340,370,840,1350
260 CLS:
   PRINT"            < PAYMENT AMOUNT >":
   GOSUB580:GOSUB610:GOSUB620
270 PM=PV*I/(1-(1+I)^-N)
280 PM=INT((PM+.005)*100)/100
290 GOTO660
300 CLS:
   PRINT"            < PRESENT AMOUNT >":
   GOSUB580:GOSUB610:GOSUB640
310 PV=PM*(1-(1+I)^-N)/I
320 PV=INT((PV+.005)*100)/100
330 GOTO660
340 CLS:
  PRINT"         < NUMBER OF PAYMENTS >"
    :GOSUB580:GOSUB620:GOSUB640
350 N=-(LOG(1-(I*PV/PM)))/LOG(1+I)
360 GOTO660
370 CLS:
 PRINT"       < AMORTIZATION SCHEDULE >"
    :GOSUB580:GOSUB620:GOSUB640
380 N=0:TI=0:BA=PV:PP=0:PI=0:
    B$="$$####,###.##":J=0:
    GOSUB780
390 N=N+1:PI=I*BA:PP=PM-PI:TI=TI+PI
400 IFPP>BAGOTO470
410 BA=BA-PP
420 J=J+1
430 IF J>5 GOTO460
440 GOSUB790
450 GOTO390
460 INPUT
    "       TO CONTINUE, PRESS 'ENTER'";
    X:J=0:GOSUB780:GOTO390
470 PP=BA
480 BA=0
490 J=J+1
500 IF J>5 GOTO460
510 N=N-1
520 INPUT
    "       TO CONTINUE, PRESS 'ENTER'";
    X:GOSUB660
530 INPUT
    "       TO CONTINUE, PRESS 'ENTER'";
    X:GOSUB780
540 IF INKEY$="" GOTO540
550 CLS:PRINT@256,"TOTAL INTEREST IS";
560 PRINTUSINGA$;TI
570 GOTO220
580 PRINT"ANNUAL INTEREST RATE =";
    AI;"%";:INPUT AI
590 PRINT"PAYMENTS PER ANNUM =";AP;:
    INPUT AP
600 I=AI*.01/AP:RETURN
610 PRINT"NUMBER OF YEARS =";NR;:
    INPUT NR: N=NR*AP: RETURN
620 PRINT"PRESENT VALUE = $";
630 PRINT TAB(20) USING A$;PV;:
    INPUT PV: RETURN
640 PRINT"AMOUNT OF PAYMENT =$";
650 PRINTTAB(20)USINGA$;PM;:
    INPUT PM: RETURN
660 CLS:PRINT:PRINT"LOAN VALUE";
670 PRINTTAB(25)USINGA$;PV
680 PRINT
    "INTEREST RATE                    ";
    AI;"%"
690 PRINT"PAYMENT AMOUNT";
700 PRINTTAB(25)USINGA$;PM
710 PRINT"TOTAL INTEREST";
720 PRINTTAB(25)USINGA$;N*PM-PV
730 PRINT"TOTAL PAYMENTS";
```

```
740 PRINTTAB(25)USINGA$;N*PM
750 PRINT"NUMBER OF PAYMENTS";
760 PRINTTAB(30)USING"#####";N
770 GOTO220
780 CLS:PRINT
 "PERIOD PRINCIPAL   INTEREST   BALANCE
790 PRINTUSING"###";N,
800 PRINTUSINGB$;PP,PI,BA:RETURN
810 '* AMORTIZATION SCHEDULE PRINTOUT *
        AI = ANNUAL INTEREST RATE
        AP = PAYMENTS PER ANNUM
        PV = PRESENT VALUE
        PM = PAYMENT AMOUNT
820 '   PI = PAYMENT INTEREST RATE
        IP = PAYMENT INTEREST
        BA = BALANCE
        PP = PAYMENT PRINCIPAL
         K = PAGE COUNTER
        CI = CUM. INTEREST
830 '    L = PAYMENT NUMBER
        CP = CUM. PRINCIPAL
840 CLS:PRINT
"   <  AMORTIZATION SCHEDULE PRINTOUT >
850 PRINT"ANNUAL INTEREST RATE =";AI;
    "%";: INPUT AI
860 PRINT"PAYMENTS PER ANNUM =";AP;:
    INPUT AP
870 PRINT"PRESENT VALUE = $";
880 PRINT USING A$;PV;: INPUT PV
890 PRINT"AMOUNT OF PAYMENT = $";
900 PRINT USING A$;PM;: INPUT PM
910 PRINT
    "       PRESS <C> TO CORRECT DATA
             * * TURN ON PRINTER * *
920 C$=INKEY$: IF C$="" GOTO920
930 IF C$="C" GOTO840
940 L=0: IP=0: PP=0: BA=PV: CI=0: CP=0:
    K=0: PI=AI/AP/100
950 Z$="$$#####,###.##"
960 LPRINT          "
    AMORTIZATION SCHEDULE
970 LPRINT "-------------------------
-----------------------------------
-----
980 LPRINT "LOAN AMOUNT";:
    LPRINT USING Z$;PV;
990 LPRINT      "
    ";AI;"% INTEREST
1000 LPRINT "PAYMENT     ";:
    LPRINT USING Z$;PM;
1010 LPRINT      "
     ";AP;"PAYMENTS PER YEAR
1020 LPRINT "-------------------------
-----------------------------------
------
1030 LPRINT
1040 LPRINT
    "PAYMENT   INTEREST     PRINCIPAL
            CUMULATIVE   CUMULATIVE
1050 LPRINT
      "NUMBER    PAYMENT       PAYMENT
  BALANCE     INTEREST    PRINCIPAL
1060 LPRINT
1070 LPRINT USING"####";L;
1080 LPRINT USING Z$;IP;
1090 LPRINT USING Z$;PP;
1100 LPRINT USING Z$;BA;
1110 LPRINT USING Z$;CI;
1120 LPRINT USING Z$;CP
1130 IF BA<>0 GOTO1160
1140 LPRINT CHR$(140);
1150 GOTO200
1160 IF PM<BA GOTO1240
1170 L=L+1
1180 IP=INT((100*PI*BA)+.5)/100
1190 PP=BA
1200 BA=0
1210 CI=CI+IP
1220 CP=CP+PP
1230 GOTO1070
1240 L=L+1
1250 IP=INT((100*PI*BA)+.5)/100
1260 PP=PM-IP
1270 BA=BA-PP
1280 CI=CI+IP
1290 CP=CP+PP
1300 K=K+1
1310 IFK<48 GOTO1070
1320 K=0
1330 LPRINT CHR$(140);
1340 GOTO960
1350 IT=0:N=0
1360 CLS
1370 PRINT
"<< CREDIT CARD PAYOFF APPROXIMATION >>
1380 PRINT
1390 INPUT"ANNUAL INTEREST RATE";AI
1400 I=AI/12     'MONTHLY INTEREST RATE
1410 INPUT"BALANCE DUE";B
1420 MI=(B*I)/100          'MONTH INTEREST
1430 BI=B+MI               'MONTH BALANCE +
                            INTEREST
1440 P=.05*BI              'PAYMENT
1450 B=BI-P
1460 IT=IT+MI:N=N+1
1470 IF B>10 GOTO1420
1480 PRINT"TOTAL INTEREST =";
1490 PRINT USING A$;IT
1500 PRINT"NUMBER OF PAYMENTS =  ";N
1510 GOTO220
1520 END
```

Listing 2:

```
100 '*********************************
110 '*      FINANCIAL PROGRAMS       *
     *********************************
200 CLS:
    PRINT@13,"MORTGAGE LOAN":
    A$="$$######.##"
```

```
210 PRINT"FOR PAYMENT AMOUNT
  PRESS 1 FOR PRESENT VALUE
  PRESS 2 FOR NUMBER OF PAYMENTS
  PRESS 3 FOR AMORTIZATION SCHEDULE
  PRESS 4
220 S$=INKEY$: IF S$="" GOTO220
230 S=VAL (S$)
240 IFS<1ORS>4 GOTO200
250 ONSGOTO260,300,340,370,840,1350
260 CLS:
   PRINT"            < PAYMENT AMOUNT >":
   GOSUB580:GOSUB610:GOSUB620
270 PM=PV*I/(1-(1+I)^-N)
280 PM=INT((PM+.005)*100)/100
290 GOTO660
300 CLS:
   PRINT"            < PRESENT AMOUNT >":
   GOSUB580:GOSUB610:GOSUB640
310 PV=PM*(1-(1+I)^-N)/I
320 PV=INT((PV+.005)*100)/100
330 GOTO660
340 CLS:
  PRINT"          < NUMBER OF PAYMENTS >"
    :GOSUB580:GOSUB620:GOSUB640
350 N=-(LOG(1-(I*PV/PM)))/LOG(1+I)
360 GOTO660
370 CLS:
 PRINT"       < AMORTIZATION SCHEDULE >"
    :GOSUB580:GOSUB620:GOSUB640
380 N=0:TI=0:BA=PV:PP=0:PI=0:
    B$="$$####,###.##":J=0:
    GOSUB780
390 N=N+1:PI=I*BA:PP=PM-PI:TI=TI+PI
400 IFPP>BAGOTO470
410 BA=BA-PP
420 J=J+1
430 IF J>5 GOTO460
440 GOSUB790
450 GOTO390
460 INPUT
    "       TO CONTINUE, PRESS 'ENTER'";
    X:J=0:GOSUB780:GOTO390
470 PP=BA
480 BA=0
490 J=J+1
500 IF J>5 GOTO460
510 N=N-1
520 INPUT
    "       TO CONTINUE, PRESS 'ENTER'";
    X:GOSUB660
530 INPUT
    "       TO CONTINUE, PRESS 'ENTER'";
    X:GOSUB780
540 IF INKEY$="" GOTO540
550 CLS:PRINT@256,"TOTAL INTEREST IS";
560 PRINTUSINGA$;TI
570 GOTO220
580 PRINT"ANNUAL INTEREST RATE =";
    AI;"%";:INPUT AI
590 PRINT"PAYMENTS PER ANNUM =";AP;:
    INPUT AP
600 I=AI*.01/AP:RETURN
610 PRINT"NUMBER OF YEARS =";NR;:
    INPUT NR: N=NR*AP: RETURN
620 PRINT"PRESENT VALUE = $";
630 PRINT TAB(20) USING A$;PV;:
    INPUT PV: RETURN
640 PRINT"AMOUNT OF PAYMENT =$";
650 PRINTTAB(20)USINGA$;PM;:
    INPUT PM: RETURN
660 CLS:PRINT:PRINT"LOAN VALUE";
670 PRINTTAB(25)USINGA$;PV
680 PRINT
    "INTEREST RATE                    ";
    AI;"%"
690 PRINT"PAYMENT AMOUNT";
700 PRINTTAB(25)USINGA$;PM
710 PRINT"TOTAL INTEREST";
720 PRINTTAB(25)USINGA$;N*PM-PV
730 PRINT"TOTAL PAYMENTS";
740 PRINTTAB(25)USINGA$;N*PM
750 PRINT"NUMBER OF PAYMENTS";
760 PRINTTAB(30)USING"#####";N
770 GOTO220
780 CLS:PRINT
 "PERIOD PRINCIPAL   INTEREST   BALANCE
790 PRINTUSING"###";N,
800 PRINTUSINGB$;PP,PI,BA:RETURN
1520 END
```

PCM

How To Make Your RAM Files Invisible

See PCM 4/84 P.6

By Ronald Paludan

You may have some programs and text files stored in your Model 100 which you would like to hide from inquiring eyes. Here is a program which will make filenames invisible to the menu screen and FILES command.

To use the program, RUN it and enter the filename or program name which you would like to hide. Be certain to include the .DO or .BA extension. If the file does not exist in the computer, you will receive a "FILE NOT FOUND" message. If the file exists, it will become invisible and the computer will return to the menu. An invisible text file can be accessed by entering the TEXT program and entering its name in response to: "File to edit?" An invisible BASIC program can be loaded either by entering BASIC and typing LOAD"(filename)" or by typing FILENAME.BA after SELECT: on the menu screen. To make an invisible text file or program visible again, simply run the program again and enter the filename.

An invisible file can be killed from BASIC (providing you know the filename). It also uses the same amount of memory as it would when visible. Having invisible files will not increase the number of files that the Model 100 can have in RAM.

How It Works

The Model 100's directory begins at location 63842 (decimal) and ends at location 64139. Each program and file in the Model 100 has an 11 byte entry in the directory. The first byte, called the "Directory Flag," determines the type of file, the second and third bytes are pointers to the file's location and the remaining bytes contain the filename. The program first searches the directory for the requested filename and then alters the Directory Flag by changing the status of the third bit using the XOR keyword (line 120). If the third bit equals zero, it is changed to a one which makes the file invisible. If the third bit is equal to one, the opposite occurs.

I should make two cautionary notes. First, the security provided by this program is not absolute. It is still possible for someone to find the filename of an invisible file by PEEKing the directory. Second, before experimenting with this program, save all important files and programs onto tape. While experimenting I found several interesting ways to destroy files and create cold starts by poking around the directory. I've tested this program extensively, but an error in typing it in could result in unfortunate consequences.

The listing:

```
10 REM HIDE A FILE by Ron Paludan
20 CLEAR:CLS:FILES
30 INPUT"FILENAME";F$:N$=F$
40 E$=RIGHT$(F$,3):IFE$<>".DO"ANDE$<>".B
A"THENBEEP:PRINT"Include .BA or .DO exte
nsion.":GOTO30
50 E$=RIGHT$(E$,2):F$=LEFT$(F$,LEN(F$)-3
):IFLEN(F$)>6THENF$=LEFT$(F$,6)
60 IFLEN(F$)<6THENF$=F$+" ":GOTO60
70 F$=F$+E$
80 FORX=63933TO64131STEP11:K=1:FORY=1TO6
90 IFMID$(F$,Y,1)<>CHR$(PEEK(X+Y-1))THEN
K=0
100 NEXTY:IFK=0THEN140
110 T=X-3
120 POKET,PEEK(T)XOR8
130 MENU
140 NEXTX:PRINT:PRINT"FILE NOT FOUND."
```

(Ronald Paludan is a freelance programmer living in Tucson, Arizona.)

Let Your Portable List Those Variables

By Larry Randall

Are you still letting your computer rest while you do the heavy work? On many occasions over the last several years I have wanted to add some code to, or make major changes to, an existing program. On more than a few of those occasions I used variables that were already in use, and crashed the program. The solution to this dilemma was to find what variables were in use by what I call the "search, scribble and go blind" method. Not any more! I believe teaching the "100" to do the job is a better way.

This new way I call *Variables Lister* or *Lister* for short. *Lister* will search your program, find all the variables in use, (including arrays and all their subscripts) sort them for alphanumerical order and save them in a file without duplication.

Operation

First translate the program you wished searched into a ".DO" file, then run *Lister*. Answer the question on the screen with the name of the file you just turned into text. Don't type the .DO, however. If everything is running okay the word SEARCH will be displayed on the screen. It will flicker, and you will hear a chirp with each new line searched. Search mode will end with two beeps and SORTING will be displayed. Sort mode will end with two beeps and COMPLETED will be displayed. All the variables will be stored in VLIST.

(Larry Randall, who has been using computers since 1975, is a retired Navy technician and writes software both for fun and for his job with Burroughs Corp.)

Timing

I used a 6184 byte calc program to test timing. It took 1 minute and 51 seconds to complete the task. The test program had 174 valid equates of which 121 made it to the sorter. After deleting the remaining duplicates there are 52 individual variables. Considerably faster and more efficient than the old "SSGB" method. Because of time I have not included in the program a search for "LET." (Modern programmers don't use LET anymore anyway. Do we?) If you do use LET include this program line.

```
155 S3=INSTR(A$,"LET"):IFS3
>0THENS3=S3+2:IFS3>SH
ANDS3<EPTHENSH=S3
```

This increased my test time by 16 seconds. Also, the REMs add 3 seconds.

Eccentricities

There will be occasions when strange things appear as variables in the VLIST file. As an example, when searching *Lister* itself, notice line 10. "MAXFILES" will be listed as a variable, as will "INSTR(A$, " from line 80. These are rare, easily identified and edited out. For *Lister* to watch for every oddball occurrence of this type would seriously increase its run time.

Line Analysis

0-30	Initialize and open files. DIM A$ (max variables expected).
50-60	Get a line and make a noise. If EOF go to SHELL SORTER.
80	Find the first "=" in the line and place first pointer there. If none go get new line.
100-110	Chop off all statements to the left of the one containing the equals sign. Set second pointer to left edge.
130	If REM disregard and get new line.
150-190	Look for the five most common things that will appear to the left of a variable, and place second pointer to the right of it.
210	If it is an "IF" go chop it off.
230	Between pointer #1 and #2 is the variable. Put it in an array. Forget it if it is the same as the last one.
250	If space precedes variable chop it off.
270	Chop off left end and go for more.
290-350	Really fast SHELL SORT.
370-380	Save to VLIST while eliminating the remaining duplicates.

The listing:

```
 0 REM   VARIABLES LISTER v2.1
 1 REM   by:     LARRY RANDALL
 2 REM           12306 TERI DR.
 3 REM           POWAY, CA. 92064
 10 CLEAR2000:DEFINTA-Z:MAXFILES=2:DIMA$(
 300)
 20 CLS:PRINT"VARIABLES LISTER":PRINT:INP
 UT"NAME OF '.DO' FILE";FN$:IFLEN(FN$)<10
 RLEN(FN$)>6THEN20
 30 OPENFN$+".DO"FORINPUTAS1:OPEN"VLIST.D
 O"FOROUTPUTAS2:CLS
 40 REM ---- GET LINE ----
 50 IFEOF(1)THEN290ELSE:LINEINPUT#1,A$
 60 PRINT@95,CHR$(27)+"K";"SEARCH":SOUND1
 000,1
 70 REM ---- FIND EQUATE ----
 80 EP=INSTR(A$,"="):IFEP=0THEN50
 90 REM ---- CHOP OFF LEFT END ----
 100 S1=INSTR(A$,":"):IFS1>0ANDS1<EPTHENA
 $=RIGHT$(A$,LEN(A$)-S1):GOTO80
 110 IFS1>SHANDS1<EPTHENSH=S1
 120 REM ---- DISREGARD REM ----
 130 S0=INSTR(A$,"REM"):IFS0>0ANDS0<EPTHE
 N50
 140 REM---- FIND VARIABLE LEFT EDGE ----
 150 S7=INSTR(A$,"FOR"):IFS7>0THENS7=S7+2
 :IFS7>SHANDS7<EPTHENSH=S7
 160 S2=INSTR(A$,"IF"):IFS2>0THENS2=S2+1:
 IFS2>SHANDS2<EPTHENSH=S2
 170 S4=INSTR(A$,"ELSE"):IFS4>0THENS4=S4+
 3:IFS4>SHANDS4<EPTHENSH=S4
 180 S6=INSTR(A$,"THEN"):IFS6>0THENS6=S6+
 3:IFS6>SHANDS6<EPTHENSH=S6
 190 S5=INSTR(A$," "):IFS5>SHANDS5<EPTHEN
 SH=S5
 200 REM ---- DISREGARD 'IF' ----
 210 IFS2=SHANDS2>0THEN270
 220 REM ---- DON'T SAVE DUPES ----
 230 A$(K+1)=MID$(A$,SH+1,EP-SH-1):IFA$(K
 +1)<>A$(K)THENK=K+1 ELSE270
 240 REM ---- CHOP OFF SPACE ----
 250 IFLEFT$(A$(K),1)=" "THENA$(K)=RIGHT$
 (A$(K),LEN(A$(K))-1)
 260 REM ---- CHOP OFF LEFT OF '=' ----
 270 SH=0:A$=RIGHT$(A$,LEN(A$)-EP):GOTO80
 280 REM ---- SHELL SORT ----
 290 Z5=K:BEEP:PRINT@95,"SORTING":BEEP
 300 Z5=INT(Z5/2):IFZ5=0THEN370
 310 Z2=1:Z3=K-Z5
 320 Z1=Z2
 330 Z4=Z1+Z5:IFA$(Z1)<=A$(Z4)THEN350
 340 A$=A$(Z1):A$(Z1)=A$(Z4):A$(Z4)=A$:Z1
 =Z1-Z5:IFZ1=>1THEN330
 350 Z2=Z2+1:IFZ2>Z3THEN300ELSE320
 360 REM ---- SAVE (NO DUPLICATION) ----
 370 PRINT#2,"VARIABLES FOR ";FN$:FORI=1T
 OK:IFA$(I)<>A$(I+1)THENPRINT#2,A$(I);" "
 ;
 380 NEXT:CLOSE:BEEP:PRINT@95,"COMPLETED"
 :BEEP
```

PCM

Examining Your Machine's Capabilities

By Willis Rollins

Many of us buy our computers and immediately sit down, plug them in, and start typing and fumbling our way through. If you're of this type then you might as well read on.

One of the most powerful features of the Model 100 is the built in text software. Many of us have completely overlooked the real power of this software either because we're not into writing text, or we just use it enough to transfer our text files to a larger computer with detailed word processing ability. One of the functions that the text software does best is in editing BASIC programs. Few of us have realized how powerful this editing capability really is. If you've noticed in some of the other magazines, there have been programs for editing your BASIC programs and merging BASIC programs. Why have programs to do this when the text software will do it rapidly and easily.

Let's examine a couple of the capabilities. If we have two BASIC programs that we would like to merge, we can RUN and SHIFT BREAK the first one. Saving the program as ASCII is the next step. Simply SAVE"RAM:NAME",A. From this point you have two ways to merge. The first is to RUN and SHIFT BREAK the second program, then type MERGE"RAM:NAME.DO" (the program ASCII saved as a DO file above). The disadvantage of this type of merging is that duplicate line numbers are changed to the incoming file line contents. One definite advantage is that of speed.

The second method of merging two programs is after saving the first as an ASCII file with the .DO extension, RUN and SHIFT BREAK the second program. Transfer the program to text storage by the EDIT command. Provided you have enough room in memory, use the SELECT function key (F7) at the begining of the program and the CUT function key (F6) at the end of the program. This will transfer the program to the paste buffer if enough memory is available. Return to the main menu and enter the previously ASCII-saved .DO file. With the first file in the text mode, move the cursor to the end of the program and use the PASTE key to return the contents of the paste buffer to the text file. You now have a merged text file version of the two original BASIC programs. At this point, if duplicate line numbers are present, simply renumber using the deletion and insertion text modes. When finished editing, return to the main menu and to BASIC. Type LOAD"RAM:NAME.DO." It may take a few seconds for a large file to be loaded as it is converted from ASCII to binary. Then RUN. Though this method seems to take a little longer, it has the advantage of not deleting any of the line numbers. Just keep in mind that the memory requirements are twice as large as the second program which is transferred to the PASTE buffer.

Want to delete a lot of line numbers, say 10 lines numbered from 200 to 300 in your BASIC program? It's easy if you transfer those lines using the EDIT feature to the text mode. Using a technique similar to the above, function key (F7) is used at the start of the to-be-edited program and the cut key (F6) is used at the end of the transferred part of the program. This moves that part to the paste buffer. Return to the BASIC program using the F8 key. Your 10 lines are now stored in the paste buffer and are no longer a part of the BASIC program. If you decide that the removed parts are necessary after all, return to the EDIT text mode, use the PASTE key, and return to the BASIC program using the F8 key. This will work provided you have not used the paste buffer in the interim.

Don't be scared to try out the edit and text mode features with your BASIC programs. You will find very little need for many of the editing, merge, and renumber-type programs you see. Yes, they do have an advantage here and there, but let the 100 do it.

PCM

Ask your dealer for the Porta™ series from Skyline—to help you work faster and better!

PortaCalc™ — Model 100 "electronic spreadsheet" with 14 column by 26 row workspace! Full use of the built-in function keys to save, load, screen print, report print, or to look behind the data at the formulas in use.

Full arithmetic operators including exponentiation, absolute value, integer, summation, and averaging. Calculations are done to 14 digits of precision and displayed up to 9 digits. User selectable decimal place from none through seven—not just one, two, or floating! Formula replication is included to allow fast creation of worksheets.

Worksheets may be saved, loaded, or merged using the computer's memory or cassette. Often used templates can be saved in memory for instant recall.

Two powerful utility programs are included with **PortaCalc** at no extra charge! **PortaDex**™ is a data exchange program that allows reformatting **PortaCalc** files into the DIF format used by **VisiCalc**. **PortaPrint**™ is an enhancement to the Model 100's text editor, adding the ability to adjust left, right, and top margins, and page length. Page numbering, headers, centered lines, flush right justification, new page control, and more.

Comes with extensive documentation in padded 3-ring binder. Includes tutorial, detailed reference section, and executive level sample templates. Fully illustrated with screen prints, examples, and hints.

PortaStat™ — Correlation, regression analysis, and descriptive statistics package, designed to be a working tool for today's businessman. Data can be input from cassette, memory, or keyboard. Interactive with **PortaCalc** data files! Thoroughly documented, includes tutorial.

PortaFin™ —Present value, net present value, future value, annuities, interest factors, loan constants, internal rate of return and more! This user-friendly package prompts you for the needed input data, then gives you the answer to your loan or investment problem! Interactive with **PortaCalc** files, allowing you to "read" cash flow data from memory. Keyboard or cassette input also available.

PortaMax™ — The powerful 'simplex' method of linear programming. Find the optimum 'mix' for a given set of constraints. Interacts with **PortaCalc** files to avoid time-wasting repetitive data entry. Tutorial, examples, and bibliography included.

PortaFolio™ — Stock and bond valuation, betas, Macaulay's duration, yield to maturity, bond swap calculations, and much more. All in an easy to use program, interactive with **PortaCalc** data files. Supplied with comprehensive instructions and reference manual.

PortaCalc

All **Porta** series programs are supplied on cassette and require 24K RAM.
Pricing — PortaCalc is **$69.95,** all others are **$44.95** each.
If ordering direct, add $2.00 shipping and handling per order.
Dealer inquires invited.

VISA MasterCard

Model 100 is a trademark of Tandy Corp. VisiCalc is a trademark of VisiCorp. DIF is a trademark of [illegible]

Skyline MARKETING CORP

4510 W. Irving Park Rd. • Chicago, IL 60641 • (312) 286-0762

You'll DO Better With This *TEXT* Utility

By Tino Delbourgo

The TRS-80 Model 100 has very powerful word processing functions. It does have a few limitations, but because of the way that PoCo stores everything as files, new word processing power can nevertheless be added to the TEXT program. As a related issue, how often have you wanted to LLIST BASIC programs to a width of your choosing? The following program is very short and performs three functions which I hope you will find useful.

The first verifies DO files which have been saved to tape. It is impossible to do this normally with the CLOAD? command; and unlike the CLOAD? command it just checks to see whether the file was saved properly — it does not compare it with what is currently in memory. Simply type in 1 from the menu and then a prompt will ask you for the filename of the program stored on tape. Just enter the name; no "CAS:" or ".DO" need be included.

The second function will justify left and right margins of the printer output of a TEXT file. It will also give your text a margin. You must, of course, have the TEXT file stored in RAM for this function to work properly. Type in 2 from the menu, then enter the margin size, printer width and the filename of the TEXT RAM file. No "RAM:" or ".DO"please!

Option 3 will LLIST a BASIC program in any printer column width. Just enter the width and the filename (no "RAM:" or ".DO") of the BASIC program. The BASIC program must have been stored in RAM and SAVEd in a SAVE"filename",A format beforehand.

(Tino Delbourgo is a 13-year-old computer hobbyist who lives in Hobart, Tasmania, Australia. He and his younger brother, Daniel, and his father, Robert Delbourgo, who is a physics professor, have written and published a number of programs for both the Model 100 and the Color Computer.)

This is a good program to have stored permanently in RAM if you use DO files a lot. Good DOing!

The listing:

```
5 CLEAR1000
7 ONERRORGOTO500
10 CLS:PRINT@2,"DOing Utilities - by Tin
o Delbourgo":PRINTSTRING$(40,"*");
20 PRINT"TYPE:":PRINTTAB(5);"1. Verify f
iles":PRINTTAB(5);"2. Justify printer ou
tput":PRINTTAB(5);"3. LLIST in any print
er width"
30 PRINT:PRINTSPACE$(8);"Type your choic
e number ";
40 A$=INPUT$(1):ONVAL(A$)GOTO60,90,420
50 GOTO40
60 CLS:LINEINPUT"Filename:";F$
65 F$="CAS:"+F$+".DO"
70 OPENF$ FOR INPUT AS1
75 PRINT"Found:";F$
80 IFEOF(1) THENCLOSE:END
85 LINEINPUT#1,F$
88 GOTO80
90 CLS:INPUT"Margin size";J:INPUT"Text w
idth";K
95 LINEINPUT"Filename:";F$:F$=F$+".DO"
100 H=0:B$="":LPRINT:LPRINTTAB(J);:OPENF
$FORINPUTAS1
110 IFEOF(1)THENLPRINT:CLOSE:END
120 A$=INPUT$(1,1)
125 IFA$=CHR$(10)THEN110
130 IFA$=CHR$(13)THENH=0:LPRINTB$:LPRINT
TAB(J);:B$="":GOTO110
140 B$=B$+A$:H=H+1:IFH<K THEN110
150 FORN=H TO1STEP-1
160 C$=MID$(B$,N,1):GOSUB400:IFER=-1THEN
NEXTN:LPRINTB$:H=0:B$="":LPRINTTAB(J);:G
OTO110
170 S$=LEFT$(B$,N)
180 IFRIGHT$(S$,1)=" "THENS$=LEFT$(S$,LE
N(S$)-1):GOTO180
190 IFLEN(S$)=K THEN300
200 IFLEFT$(S$,1)=" "THEN250
210 U=1
220 GOSUB390:IFER=0THENS$=LEFT$(S$,U-1)+
" "+RIGHT$(S$,LEN(S$)-U+1):U=U+1
225 IFLEN(S$)=K THEN300
230 U=U+1:IFU>LEN(S$)THEN210
240 GOTO220
250 U=LEN(S$)
260 GOSUB390:IFER=0THENS$=LEFT$(S$,U-1)+
" "+RIGHT$(S$,LEN(S$)-U+1):U=U-1
265 IFLEN(S$)=K THEN300
270 U=U-1:IFU<1THEN250
280 GOTO260
300 LPRINTS$:B$=RIGHT$(B$,LEN(B$)-N)
310 IFINKEY$=CHR$(27)THENCLOSE:END
320 LPRINTTAB(J);:H=LEN(B$):GOTO110
390 C$=MID$(S$,U,1)
400 IFC$=" "ORC$="."ORC$="?"ORC$="!"ORC$
=";"ORC$=","ORC$=":"ORC$=CHR$(34)THENER=
0ELSEER=-1
410 RETURN
420 CLS:INPUT"Printer width";W
430 LINEINPUT"Filename:";F$
440 F$=F$+".DO"
450 OPENF$FORINPUTAS1
455 B$=""
460 IFEOF(1)THENLPRINTB$:CLOSE:END
470 A$=INPUT$(1,1):IFA$=CHR$(13)THENLPRI
NTB$:GOTO455
480 IFA$=CHR$(10)THEN460
490 B$=B$+A$:IFLEN(B$)=W THENLPRINTB$:GO
TO455
495 GOTO460
500 IFERR<>18THENERRORERRELSEPRINT"Verif
y failed":CLOSE:END
```

Back Issue Availability

All back issues sell for the single issue cover price of $3. In addition, there is a $1.50 charge *per issue* for postage and handling within the United States, Canada and Mexico.

VISA, MasterCard, and American Express accepted. Kentucky residents please add 5 percent state sales tax.

HARDWARE

New Friction Option For Printers

When I was deciding to buy a printer, one of the most important aspects that I had to consider was whether I wanted the "Friction" option. All printers have the usual "Tractor" mechanism, and if you desire the added convenience of using single sheets of paper, such as stationery, your personal letterhead or whatever, the cost is usually around $100 extra.

Now, for those of you who have bitten your fingernails away completely because you now wish you paid the extra scratch for the friction feature, a cheaper (but sturdy) way to get the same results is available for *any* printer. It's called the Paper Tractor.

The Paper Tractor is a flexible plastic device that is inserted into your tractor mechanism and will carry your single sheets of paper through without any hassle.

To use this handy device, you just place the paper you would like to use in and under the half-inch folds provided on the top, right and left hand sides of the "tractor." The fit is snug and the paper will stay firmly in place. Then just thread it over your printer sprockets as though it was your usual tractor-feed paper.

You may align it at this point, checking to make sure that your print head is at the desired height on the paper. Then just print as normal.

One precaution you should take while printing is to make sure that you do not go past the end of the paper and onto the device itself, or worse, onto the platen of your printer. If your printer has a "paper out" detector, it will still function as usual.

Even though my printer has friction feed as a standard feature, I tried the Paper Tractor as described and it worked beautifully. The samples I ran were at 9600 Baud and bi-directional, which I thought might have had a negative effect because of the speed and irritation to the Paper Tractor, but it came out beautifully.

After using the tractor a couple of times, I found that I could even print on the very last line of my samples since the Paper Tractor itself extends into the paper-out switch because of its length.

The only inconvenience I can imagine is that if you intend to print an awful lot, the time needed to withdraw the Paper Tractor and insert new sheets of paper will slow things up a bit.

The Paper Tractor is a handy complement for you non-friction printer owners. It handles up to 11 x 14 paper, will work with any printer, and can be used instantly by anyone. The documentation provided does not actually explain how to use it, but if after looking at the photos provided you can't figure it out, you shouldn't be near a computer, anyway.

(Paper Tractor Ltd., 1 South Fairview, Goleta, CA 93117, $11.95)

—Steve Schechter

SOFTWARE

Financial Analysis On The Model 100

How many times have you wondered whether the yield quoted to you on a bond, a CD, or annuity was really accurate or even in the ballpark? Have you been told that the interest rate on a loan is only eight percent, only to find out later that it's really 15.4 percent? If you engage in financial transactions such as amortizations, interest, annuities, or bond analysis, then this financial analysis package from York Software may be for you.

Financial Analysis Package #1 comes on a cassette with a 16K version on one side and the various programs separated into small enough packages to run on an 8K machine. The 16K version uses approximately 11K plus overhead. If you have programs you really like to leave in the computer, then use the 8K versions. In a couple of instances the 8K programs seemed to run slightly faster than the combined 16K version. There are four programs: amortization, compound interest, annuities, and bond analysis. In the 16K version, these programs are accessed from a menu. For the 8K programs, you have to load each separately. What can these do for you?

The well-documented manual of 23 pages steps you through each analysis with examples. For the amortization program, one can calculate the equal payment amount, the original loan balance, the number of payments, and the interest rate when supplied with menu requested information. Each menu has the maximum number of spaces (indicated by nines) for each entry. If you are entering, for instance, the principal amount, you are given 999999.99 to fill up. To enter 100.00, you must type 000100.00. Counting the necessary zeros is a distraction if you are trying to hurry through. If you make a mistake in entering the data, the program traps the errors, resets the data entry menu, and awaits your re-entry of the data. There is a soft beep to audibly notify you.

The compound interest program will determine future values, present values, and compare interest rates when the frequency of compounding changes. I entered last year's IRA contribution and plugged in 8.5 percent dividend for many years, as I am still young. It took approximately 3½ minutes to compute the future value for 26 years, compounding daily. The manual forewarns that some calculations may take a long time since the programs are written exclusively in BASIC. What appears to be time consuming is two calculations; the first, the actual number of days, including leap years are calculated, and second, a rather large loop to compute dividends when compounding daily. If you purchase this package and decide to use a long time period, I suggest you select compounding on a quarterly, semi-annually, or yearly basis to speed up the calculations. The loss in accuracy is barely noticeable.

The third program is for calculating the final value or the amount to be deposited on some frequency schedule if you're interested in annuities. Again the information needed to solve the problems is entered through a menu driven screen.

The last program determines the yield and the purchase amount of bonds. This program differs in its calculations from the compound interest program in that bonds are usually purchased at a discount or at a premium returning in some future time a guaranteed amount. This program selection takes the misery out of trying to figure it with a calculator.

The financial analysis package should be of interest to real estate salespersons who often need to know exactly what the monthly payments are without resorting to the tables broken down to the nearest $1000 and ¼ percent interest rate. The amortization program also works very well where consumer credit information is readily needed, such as used and new car dealers. If you are making investments in the bond market, this package of programs will give you figures to compare with the ones supplied by your broker.

If you're involved with any of the financial analysis programs available in the package, you will find it worth your time to consider this package. You may find it a little slow running in the compound interest section, but for a BASIC program, this one runs very cleanly and is very nicely menu-driven both for the programs and the data entry sections. The manual is very clear and has good examples to try. The package should be a welcome addition to the Model 100 for those needing to solve interest and dividend type problems.

(York Software, 2885 Tanglefoot Lane #8, Bettendorf, IA 52722, $39.00)

— Vincent Lord

New Products

Rechargeable Battery System

Now available for the rechargeable Model 100, NEC's PC-8201A and other 6-volt briefcase portables, is a lightweight, external battery system by Blue-

The Rackseller

ALABAMA	
Florence	Anderson News Co.
Madison	Madison Books
ALASKA	
Fairbanks	Electronic World
	Fairbanks News Agency
Kotzebue	Arctic Bookstore, Inc.
ARIZONA	
Phoenix	Road Runner
	Computer Pro
Scottsdale	Softwareland Corp.
Tucson	Anderson News Co.
CALIFORNIA	
Atwater	Electro Brain
Citrus Heights	Software Plus
Half Moon Bay	Strawflower Electronics
North Hollywood	Levity Distributors
Sunnyvale	Computer Literacy
CONNECTICUT	
Hartford	The Aetna Life Club Store
DELAWARE	
Wilmington	Normar Inc.—The Smoke Shop
FLORIDA	
Ft. Lauderdale	Software Connection
Miami	Newsrack
Panama City	Computer Systems Group
Pensacola	Anderson News Co.
Tallahassee	Anderson News Co.
Winter Park	Independent Computer Systems
GEORGIA	
Atlanta	Guild News Agency
Columbus	Muscogee News Co.
Trenton	The Phone Line
ILLINOIS	
Chicago	Kroch's & Brentano's Jackson Street
	Kroch's & Brentano's Wabash Ave.
Chicago	Prairie News Agency
Oakbrook	Kroch's & Brentano's
INDIANA	
Elkhart	Elkhart City News & Book Store
Indianapolis	A—Computer Store
Madison	Arco Office Supplies
Scottsburg	Radio Shack of Scottsburg
IOWA	
Davenport	Interstate Book Store
KANSAS	
Wichita	Amateur Radio Equipment Co.
	Lloyd's Radio
KENTUCKY	
Benton	Anderson Electronics
Pikeville	Gus-Stan Enterprises
LOUISIANA	
Slidell	Radio Shack
MAINE	
South Portland	Portland News Co.
MASSACHUSETTS	
Littleton	Computer Plus
Marlboro	Radio Shack
Woburn	Microcon Software Centers
MICHIGAN	
Novi	MI Software Dist., Inc.
Rochester	Rochester Book Center
NEBRASKA	
Lincoln	Hobby Town
NEVADA	
Las Vegas	Hurley Electronics
NEW HAMPSHIRE	
Manchester	Brookwrights
Petersborough	Radio Shack
West Lebanon	Verham News Corp.
NEW JERSEY	
Hackensack	Total Circulation Service
Marmora	Outpost Radio Shack
Midland Park	Software City
Pennsville	Dave's Electronic Radio Shack
NEW MEXICO	
Albuquerque	News and Bookstore
Albuquerque	Page One Newsstand
NEW YORK	
Brooklyn	Cromland, Inc.
Johnson City	Unicorn Electronics
New York City	Usercom System, Inc.
NORTH CAROLINA	
Cary	It's Just For You, Inc.
NORTH DAKOTA	
Fargo	Computer Associates
OHIO	
Cincinnati	Softwarehouse
Toledo	Leo's
	Reitz Electronics, Inc.
OREGON	
Eugene	Libra Books, Inc.
Medford	John's News Stand
Portland	80-Plus
Roseburg	Rainy Day Bookshop
Tigard	80-Plus of Tigard
PENNSYLVANIA	
Pleasant Hills	Pittsburgh Computer Store
SOUTH CAROLINA	
Greenville	Palmetto News Co.
North Charleston	The Green Dragon
TENNESSEE	
Antioch	P.S. Computers
Chattanooga	Anderson News Co.
Knoxville	Anderson News Co.
Memphis	Computer Center
	Software, Inc.
Nashville	Mosko's Book Store
Smyrna	Delker Electronics, Inc.
TEXAS	
Austin	Software & Things
Dallas	Micro Concepts, Inc.
Elgin	Homing Pigeon
Ft. Worth	A & A International
	Byteworks, Inc.
	RFI Electronics
Irving	Software Access
WASHINGTON	
Blaine	Compulit Dist.
Richland	C & J Electronics Computer Center
Seattle	Adams News
WISCONSIN	
Appleton	Badger Periodicals
Janesville	Book World
Milwaukee	Campus Corner
Portage	Sam's Electronics
CANADA:	
ALBERTA	
Calgary	Rainbow Software Services
Edmonton	Kelly's Software Distributors
NOVA SCOTIA	
Halifax	Atlantic News

Also available at selected B.Dalton Bookseller stores in the United States and Canada.

stem Productions.

The batteries come in three sizes to suit an individual's needs. All batteries are lightweight, palm-sized and fit into a briefcase or computer hard-case and come with an attaching 6-foot, 6-volt cable.

The Prairie Power 8-ampere-hour unit can run the Model 100 continuously for over 150 hours (our last test gave 160 hours). The 5.5 amp-hour battery supplies 100 hours of usable power. Finally, the 2.6 amp-hour unit supplies 50 hours of continuous power and is only slightly larger than a pack of cigarettes.

The Prairie Power rechargeable units range in price from $21.95 (for 54 hour charge) to $34.95 (for over 150 hours of continuous operation). The battery charger is $10.95.

Write Bluestem Productions, Box 334, Wayzata, MN 55391; or call (612) 471-7795.

8K RAM Module

Purple Computing has developed, and is delivering from stock, a new 8K RAM memory module for the Model 100 and PC-8201 computers. According to the Purple Computing people, this new module functions exactly like the manufacturers' memory modules and is designed as a replacement for those parts. The module provides 8,192 bytes of memory in a single dual inline package. You can call them for information or for placing orders at (805) 987-4788; or write to Purple Computing, 4807 Calle Alto, Camarillo, CA 93010.

Carrying Case/ Work Station

The Chip-Tote, by Kangaroo Video Products, is a carrying case for the Model 100, NEC PC-8201 and Epson HX-20 portable computers that doubles as a desk.

It features a slim, fully foam-padded design that opens up into a one-piece work station.

The stand-up utility top holds papers upright for easy reference, and inner pockets keep notebooks and notepads handy. A zippered pouch holds an AC adapter, acoustic coupler, modem cord and extra batteries.

Chip-Tote comes with adjustable shoulder strap and a hand strap. Kangaroo Video Products, Inc. specializes in designing and manufacturing carrying cases for sensitive electronic equipment.

The Chip-Tote is made of tough, durable DuPont Cordura nylon in black or smoke gray. At $59.95, it's available from Kangaroo Video Products, Inc., 9190 Manor Drive, La Mesa, CA 92041. Phone (619) 698-0230.

SAVE $120

on the TRS-80 Model 100 with 8K RAM

Model 100 Cables
CoCo to Model 100 Cables
— $19.95
CoCo to Model I, II, III, IV
— $24.95
C-10 Tapes — any quantity
49¢ each

stock #26-3801

$679

For quick delivery call:

Model 100 8K Memory
Upgrade Module

$69.95

each

SPECTRUM PROJECTS

93-15 86th Drive
P.O. Box 21272
WOODHAVEN, NY 11421
(212) 441-2807

PCM CERTIFICATION SEAL

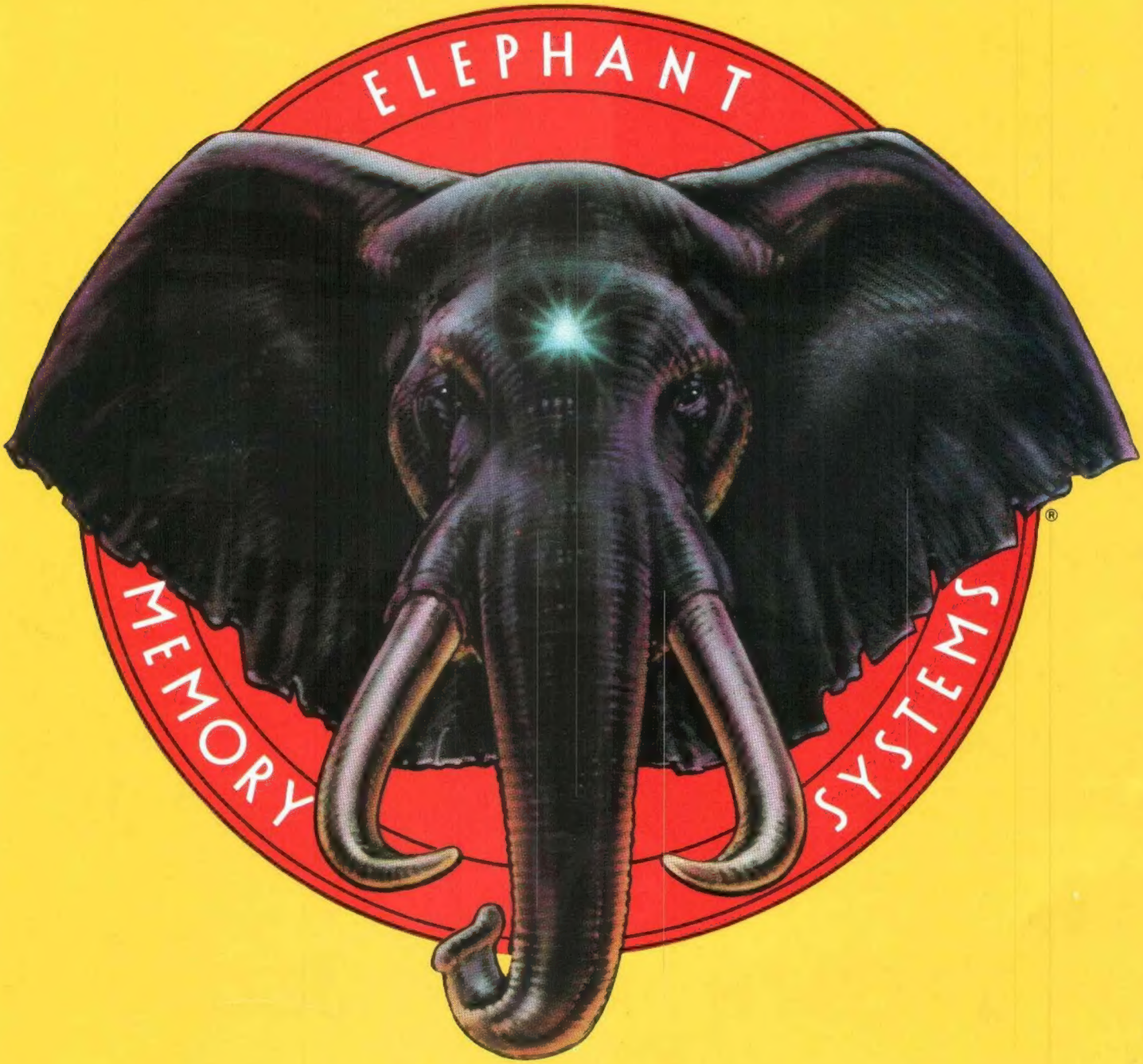
plus $5 shipping and handling
N.Y. residents add sales tax

Call our Bulletin Board for free Model 100 programs.

Now on line . . . 20 MEG. CORVUS HARD DISK DRIVE

BBS #1 — (212) 441-3755
BBS #2 — (212) 441-3766
BBS #3 — (212) 441-5719
BBS#4 - (212)441-5907

REMEMBER.



ELEPHANT™ NEVER FORGETS.

A full line of top-quality floppies, in virtually every 5¼" and 8" model, for compatibility with virtually every computer on the market. Guaranteed to meet or exceed every industry standard, certified 100% error-free and problem-free, and to maintain its quality for at least 12 million passes (or over a lifetime of heavy-duty use).

Contact Dennison Computer Supplies, Inc., 55 Providence Highway, Norwood, MA 02062 or call toll-free 1-800-343-8413. In Massachusetts, call collect (617) 769-8150. Telex 951-624.

Dennison